ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

13

Economía doméstica con el ordenador personal

José Luis Morales

SINCLAIR AMSTRAD COMMODORE IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

13

Economía doméstica con el ordenador personal

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo XIII. **Economía doméstica con el ordenador personal.**
JOSE LUIS MORALES, Profesor Mercantil.

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.º G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-045-6.
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M- 42.837-1986
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid

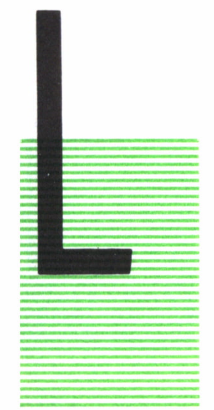
Octubre, 1986.
P.V.P. Canarias: 365,-

INDICE

Los programas que aparecen en este libro funcionan en los ordenadores:

IBM–PC, XT, AT y compatibles.
AMSTRAD–464, 664, 6128, 1512.
SINCLAIR–SPECTRUM 48 K, 128 K, PLUS, PLUS 2.
MSX–Todos los modelos.
COMMODORE–CBM 64 y CBM 128.

INTRODUCCION



LA aparición de los ordenadores personales ha sido seguida por su impresionante proliferación, debida al abaratamiento de los materiales, lo que ha traído consigo el que se vaya introduciendo en el hogar, con la posibilidad de utilizarlo, tanto para distraer los ratos de ocio jugando como para realizar otras aplicaciones de tipo práctico: escribir pequeños o grandes programas para realizar controles domésticos, etc. Si nos paramos a pensar, esto no se aplica sólo a los mayores y jóvenes, sino también a los pequeños, que poco a poco pueden realizar sus pequeñas aplicaciones.

A veces la tentación de probar y ver hasta dónde es uno capaz de ser habilidoso es sumamente emocionante. Naturalmente existe un enorme número de formas de realizar las aplicaciones en el ordenador. Nosotros, en este libro, nos vamos a referir al control doméstico y para ello utilizaremos un lenguaje sencillo para el lector.

Existen distintos lenguajes para realizar el control de las operaciones contables, como pueden ser:

Assembler
Basic
Pascal
APL

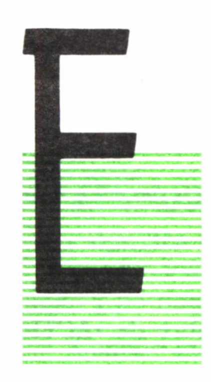
En este libro vamos a introducir ejemplos en Basic y Pascal, por ser éstos los más corrientes en los diferentes ordenadores existentes en el mercado y cuyos compiladores e intérpretes son muy asequibles.

El objeto de este libro de Contabilidad Casera o doméstica no es otro que introducir al lector poco a poco en un procedimiento que le permita llevar su control doméstico. Por ello, voy a tratar de iniciarlo con un criterio eminentemente didáctico, pensando en aquellas personas que deseen

adquirir conocimientos contables básicos, primero y sobre todo para saber el terreno que pisan.

Por consiguiente, nuestra enseñanza de esta materia comienza explicando qué es y para qué sirve la Contabilidad, partiendo del supuesto de una familia cuyo patrimonio experimenta diversas variaciones como consecuencia de las operaciones comerciales que realiza, operaciones que deben ser captadas por la Contabilidad, a través de los registros correspondientes, que se irán explicando paso a paso, de forma que el lector vaya avanzando en el conocimiento contable como por una suave rampa sin escalones de difícil ascensión.

CONTABILIDAD, PATRIMONIO E INVENTARIO 1



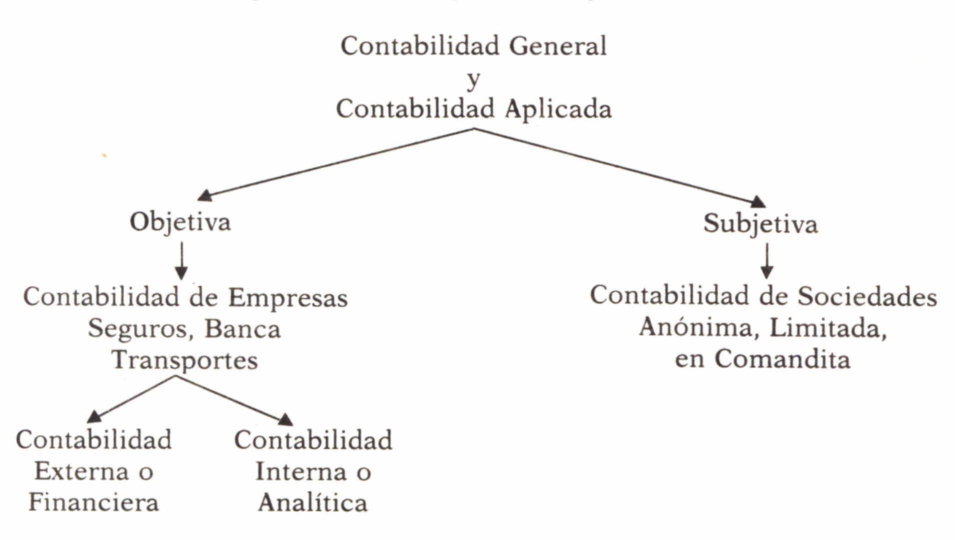
EN este primer capítulo vamos a establecer las bases de la Contabilidad, para lo cual veremos qué es y para qué sirve. Vamos, pues, a comenzar definiendo qué es Contabilidad.



CONTABILIDAD

Es la ciencia del control de los negocios. Ampliando esta definición podemos decir que el control de los negocios consiste: en registrar las variaciones que experimenta el patrimonio (en este caso de la familia) y en determinar la clase y cuantía de las ganancias o pérdidas que se obtienen.

La contabilidad puede dividirse para su mejor estudio en varias ramas:



Nosotros ahora sólo vamos a tratar de la Contabilidad General, que comprende los principios básicos, elementales y fundamentos de la Ciencia Contable.

Como modelo práctico vamos a desarrollar la contabilidad de una familia, registrando las variaciones que se producen en su «PATRIMONIO» como consecuencia de las operaciones que realiza, determinando también las ganancias o pérdidas que obtiene de acuerdo con lo expresado en la definición de Contabilidad.

Hemos hablado antes del Patrimonio y lo hemos entrecomillado. ¿Qué es el Patrimonio?



PATRIMONIO

Es el conjunto de bienes, derechos y obligaciones que pertenecen a una persona y que pueden valorarse en dinero.

El patrimonio consta de dos partes:

Una, que podemos llamar positiva y que recibe el nombre de «ACTIVO», y otra negativa, que denominamos «PASIVO».



ACTIVO

Es el conjuto de bienes y derechos que posee una persona y que puede valorarse en dinero.

Bienes. Son todos aquellos objetos y productos que, teniendo un valor en dinero, pertenecen a una persona, o sea, forman parte de su Patrimonio.

Derechos. Son créditos o deudas a favor de la persona contra otras, expresados en dinero y que, lógicamente, también forman parte de su Patrimonio.



PASIVO

Es el conjunto de obligaciones a cargo de la persona y a favor de otras personas, que pueden valorarse en dinero.

Obligaciones. Son deudas contraídas por la persona con otras.

La diferencia entre la parte positiva y la negativa del patrimonio, es decir, entre el Activo y el Pasivo, recibe el nombre de:

Capital Líquido o Capital Neto

o también

Patrimonio Líquido o Neto

Al comparar el ACTIVO con el PASIVO pueden darse los siguientes casos:

a) Existe Activo, pero no existe Pasivo. En este caso el Activo es el propio Capital. La persona no debe nada.
b) El Activo es superior al Pasivo. Esto es lo normal. La diferencia es positiva, la persona debe parte de lo que tiene.
c) El Activo es igual al Pasivo. En este caso la persona debe todo lo que tiene.
d) El Activo es inferior al Pasivo. En este caso la diferencia es negativa y significa que la persona debe más de lo que tiene.
e) No existe Activo, pero existe Pasivo. En este caso la persona debe todo y no tiene nada para hacer frente a sus deudas.

Cuando una persona comienza a registrar sus operaciones con el sistema que más adelante iremos indicando, lo primero que tiene que hacer es saber con qué medios cuenta para llevar adelante su CASA, para lo cual establece el correspondiente INVENTARIO. Vamos, pues, a definir qué es un INVENTARIO.



INVENTARIO

Es la relación detallada y ordenada de su PATRIMONIO. No debe confundirse PATRIMONIO con INVENTARIO.

— El Patrimonio es un conjunto de bienes, derechos y obligaciones.
— El Inventario es la relación de esos bienes, derechos y obligaciones.

Al igual que el Patrimonio, el Inventario (que es su fiel reflejo) estará dividido en ACTIVO y PASIVO.

Desde un punto de vista formal, el Inventario consta de:

Encabezamiento
Cuerpo
y
Pie

Encabezamiento:

Consiste en una frase introductoria con el siguiente contenido: «Inventario n.º . del Capital activo, pasivo y líquido de D. ...., domiciliado en ...., efectuado en fecha ..».

Cuerpo:

Es la parte central del Inventario y, por tanto, la más importante, pues en ella se detalla el Activo, Pasivo y Capital líquido.

Pie:

Parte última del Inventario, que consiste simplemente en un epílogo, cuyo texto puede ser: «De acuerdo con el presente inventario, mi capital líquido al día de la fecha asciende a pesetas ....

Fecha y firma.

Veamos ahora, con un ejemplo práctico, cómo se forma el inventario que puede dar comienzo a nuestros registros de las distintas operaciones que vayamos haciendo día a día. Para ello, en primer término, efectuemos un examen de los bienes, derechos y obligaciones que tenemos, es decir, el patrimonio, y les asignaremos los nombres que más usualmente reciben los apuntes contables de dichos bienes, obligaciones y derechos.

Supongamos una familia que tiene:

Bienes	**Nombre contable**
50.000 ptas., en dinero efectivo.	Caja.
200.000 ptas., depositadas en Banco.	Banco.
60.000 ptas., en objetos vendibles.	Mercaderías.
120.000 ptas., valor de un mueble.	Mobiliario y enseres.
840.000 ptas., un vehículo.	Elementos de transporte.
6.450.000 ptas., valor de un piso en propiedad.	Edificio e inmuebles.
DERECHOS	
45.000 ptas., en recibos por cobrar.	Efectos comerciales.
24.000 ptas., le adeudan por ventas.	Clientes.
15.400 ptas., le adeudan.	Deudores.

OBLIGACIONES	
50.000 ptas., en recibos pendientes de pagar.	Efectos a pagar.
20.000 ptas., debe por compras.	Proveedores.
780.000 ptas., debe de otros conceptos.	Acreedores varios.

Ahora con estos datos podemos efectuar el Inventario correspondiente, pero antes hay que hacer las advertencias siguientes:

a) Los bienes y derechos irán ordenados de mayor a menor disponibilidad, entendiéndose por disponibilidad el grado de facilidad que tienen para realizarse en dinero.

b) Dentro de cada elemento de los que forman el inventario hay que detallar las partidas que lo componen; no basta, por ejemplo, poner CLIENTES ... 24.000, hay que detallar el nombre de cada cliente con el importe de su deuda respectiva.

Inventario núm. 1 del Capital Activo, Pasivo y Líquido de la familia de esta plaza don Antonio Gosta Pérez, domiciliado en la calle de la Montera, 45, de Madrid, efectuado en el día de la fecha:

ACTIVO	Importes	
	Parciales	**Totales**
CAJA		50.000
dinero efectivo, según arqueo	50.000	
BANCOS		200.000
saldo en c/c Banco Hispano	100.000	
saldo en c/c Caja Postal	100.000	
EFECTOS COMERCIALES		45.000
recibo a cargo de Comercial X, vto. 3-4-86	45.000	
CLIENTES		24.000
Xiuz Martínez, industrial	24.000	
DEUDORES		15.400
Don Pedro Pérez	15.400	
MERCADERIAS		60.000
80 cintas de clase x, a 500 ptas.	40.000	
10 cintas de clase Y, a 2.000 ptas.	20.000	
ELEMENTOS de TRANSPORTE		840.000
coche Ford	840.000	
MOBILIARIO y ENSERES		120.000
equipo de música	120.000	
EDIFICIO e INMUEBLES		6.450.000
Chalé en Colmenar del Arroyo	4.000.000	
Piso en la calle de Bordadores	2.450.000	
Total activo	7.804.400	7.804.400

PASIVO	Importes	
	Parciales	Totales
EFECTOS A PAGAR		50.000
recibo girado por Comercial y vto. 3-5-86	50.000	
PROVEEDORES		20.000
Piuz Garnino, industrial	20.000	
ACREEDORES VARIOS		780.000
Cedro Pérez	280.000	
Gómez Caso	500.000	
Total Pasivo	850.000	850.000

RESUMEN

Total Activo	7.804.400
Total Pasivo	850.000
Capital Líquido	6.954.400

De acuerdo con el presente inventario, el capital líquido al día de la fecha asciende a 6.954.400 pesetas.

Madrid, 10 de noviembre de 1986

Antonio Costa Pérez

Hemos visto un modelo de Inventario con todo género de detalles, como debe hacerse en la vida real; no obstante, para la práctica de los ejercicios contables, y con objeto de facilitar su comprensión, suele emplearse una forma resumida de Inventario en la que sólo constan los elementos que componen el Activo y el Pasivo, así como la diferencia entre ambos, el Capital líquido, sin detallar las partidas.

Con objeto de acostumbrarse a esta forma resumida de Inventario, a continuación vamos a repetir de manera extractada el inventario anterior.

ACTIVO	
CAJA	50.000
BANCOS	200.000
EFECTOS COMERCIALES	45.000
CLIENTES	24.000
DEUDORES	15.400
MERCADERIAS	60.000
ELEMENTOS de TRANSPORTE	840.000
MOBILIARIO y ENSERES	120.000
EDIFICIO e INMUEBLES	6.450.000
Total Activo	7.804.400
PASIVO	
EFECTOS A PAGAR	50.000
PROVEEDORES	20.000
ACREEDORES VARIOS	780.000
Total Pasivo	850.000

RESUMEN

Total Activo	7.804.400
Total Pasivo	850.000
Capital Líquido	6.954.400

Madrid, 10 de noviembre de 1986

Aclaremos algunos conceptos que hemos empleado en el Inventario que precede (completo y resumido), sin perjuicio de que posteriormente volvamos sobre lo mismo.

En primer lugar, Clientes y Deudores son ambas cuentas que quieren decir que nos deben dinero, pero mientras las deudas de los Clientes son por mercancías compradas que aún no han pagado, las deudas de los Deudores son por cualquier otro concepto distinto de la compraventa de mercancías.

Los Acreedores y Proveedores (a ambos les debemos dinero), pero mientras las deudas con los Proveedores son por mercancías vendidas que aún no se han pagado, las deudas a los Acreedores son por cualquier otro concepto distinto del de la compraventa de mercancías.

Otro término que ha aparecido en el inventario es el de «arqueo». El arqueo consiste en contar las monedas y billetes que poseemos para comprobar que su «existencia» coincide con los apuntes contables.

Hasta ahora hemos visto cómo se forma un Inventario, lo que es el pun-

to de partida o arranque, pero una vez se ha dado comienzo, empieza a producirse una serie de operaciones de la vida normal, como, por ejemplo:

Compras
Ventas
Cobros
Ingresos
Retirada de fondos
etc.

que tienen que ser registradas por nuestra Contabilidad (recuérdese que uno de sus objetivos es registrar todas y cada una de las variaciones que experimente nuestro Patrimonio), y que reciben el nombre de HECHO CONTABLE.

HECHOS CONTABLES 2



SON las operaciones mercantiles efectuadas por la persona, que aparecerán como un registro en su Contabilidad.

Los Hechos se pueden dividir en:

Permutables
Modificables
y
Mixtos



PERMUTABLES

Son aquéllos que producen una variación entre dos o más elementos del Patrimonio, sin que se altere su situación (Capital Líquido).

Por ejemplo: Supongamos que la persona compró un equipo de música por 120.000 pesetas y las paga al contado. Es evidente que el equipo hará aumentar su Patrimonio en su parte positiva (ACTIVO), pero también su CAJA disminuirá en la misma cifra, con lo cual se habrá producido una variación en los elementos patrimoniales del ACTIVO sin alterar su capital líquido.

Igual sucedería si en lugar de comprar cancelara parte de una deuda a través del dinero existente en el Banco. En este caso sería una disminución de nuestra cuenta bancaria y también una disminución de la deuda, con lo cual tampoco habría diferencia en nuestro Capital líquido.



MODIFICABLES

Son aquéllos que modifican la cuantía del CAPITAL líquido sin que se produzca una permutación de los elementos patrimoniales. Pueden ser aumentativos o disminutivos, segun la variación que se produce.

Por ejemplo: Supongamos que el Banco nos anuncia que abona en nuestra cuenta, en concepto de intereses, 1.000 pesetas. Nos hallaremos ante un Hecho modificativo aumentativo, ya que el elemento Banco de nuestro patrimonio aumenta sin disminuir el PASIVO, con lo cual aumenta nuestro CAPITAL.

Supongamos ahora que el equipo de música se nos rompe y la reparación nos cuesta un valor de 3.000 pesetas.

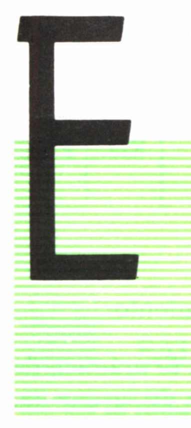
Este hecho también modifica, pero en signo contrario, disminuyendo nuestro ACTIVO, con lo cual nuestro CAPITAL también quedará disminuido.



MIXTOS

Son aquellos que a la vez tienen carácter permutable y modificable, es decir, que producen variación entre dos o más elementos patrimoniales y al mismo tiempo alteran la cuantía del Capital. Este es el caso cuando vendemos en 140.000 pesetas nuestro equipo de sonido. Esto quiere que permutamos dos elementos de nuestro Patrimonio con distinto valor, con lo cual nuestro Capital queda alterado, puesto que recibimos 20.000 pesetas más por el equipo.

LA CUENTA 3



N el capítulo anterior hemos aprendido a reflejar, mediante inventarios, las variaciones que experimenta el Patrimonio de la supuesta familia a causa de las operaciones comerciales o de la vida cotidiana.

Como fácilmente se puede suponer, sería laboriosísimo en la práctica realizar un Inventario cada vez que la familia efectúe una transacción comercial, por lo que hemos de emplear otro método para registrar los Hechos contables que se produzcan en su desenvolvimiento diario.

Este método recibe el nombre de Partida Doble y se instrumenta a través de las Cuentas.

Vamos, por tanto, a empezar definiendo qué es una CUENTA.



CUENTA

Llamamos cuenta de estado detallado de las distintas variaciones que experimenta un determinado elemento del patrimonio, o bien una determinada pérdida o ganancia.

Por tanto, para cada elemento patrimonial y para cada clase de pérdida o de ganancia abriremos y llevaremos una Cuenta, en donde registraremos las distintas variaciones que experimenta dicho elemento patrimonial, o bien las pérdidas o ganancias que se produzcan.

EL libro donde figuran todas las cuentas que necesitemos para llevar el control de nuestro hacer familiar recibe el nombre de LIBRO MAYOR.

Nosotros nos referiremos a él como MAYOR, simplemente. Una cuenta tiene un aspecto formal, esto es, que presenta dos partes llamadas DEBE

y HABER, por lo que se establece a doble folio (entradas y salidas). Cada parte consta de los siguientes componentes:

Fecha (día mes año) dd mm aa
Concepto o contrapartida
Referencia
Cantidad parcial
Cantidad total

Vamos a presentar un modelo rápido de cuenta para una mejor comprensión. En la parte superior se coloca el título de la cuenta, que es el nombre del elemento patrimonial.

Debe				Título			Haber
Fecha	Concepto	Rfa.	Ptas.	Fecha	Concepto	Rfa.	Ptas.

Rfa.: es el número de documento que produce el movimiento.

Al igual que ocurre con el Inventario, como se vio para la práctica de los ejercicios contables, se emplea un rayado simplificado de la cuenta, consistente en una especie de «T» mayúscula gigante, en la que sólo se coloca las cantidades que correspondan al DEBE y al HABER, en la parte superior el título de la cuenta y a lo sumo las letras «D» y «H» como abreviatura de las palabras Debe y Haber.

D Título	H
Cantidades del Debe	Cantidades del Haber

Cuando se pretende un ejercicio contable con todo género de detalles y, por supuesto, cuando se trata de la Contabilidad real, se emplea el modelo completo descrito en primer lugar.



ANOTACIONES

En la cuenta, como ya hemos dicho, las anotaciones en ella pueden efectuarse en el Debe o en el Haber.

— Anotar una partida o cantidad en el Debe significa:

Adeudar
Cargar
Destinar

— Anotar una partida o cantidad en el Haber significa:

Abonar
Acreditar
Originar

Gráficamente podemos representar lo anterior de la siguiente forma:

D	Título	H
Adeudar Cargar Destinar		Abonar Acreditar Originar



SALDO DE LA CUENTA

Es la diferencia entre la suma de las partidas anotadas en el Debe y la suma de las cantidades anotadas en el Haber.

— Si la suma del Debe es mayor que la del Haber, el saldo es Deudor.
— Si la suma del Haber es mayor que la del Debe, el saldo es Acreedor.
— Si la suma de las partidas del Debe es igual a la suma de las partidas del Haber, se dice que la cuenta está SALDADA.

Para registrar las partidas en las cuentas, se deben seguir los criterios que a continuacion detallamos:

En el Debe se anotarán:

Los importes iniciales y los aumentos, entradas o destinos de los elementos de ACTIVO (bienes, derechos, mobiliario, Caja...), las cancelaciones o disminuciones de los elementos PASIVOS (Obligaciones...); los Gastos y Pérdidas

En el Haber se anotarán:

Las disminuciones, salidas u orígenes de los elementos de ACTIVO (bienes, derechos, etc.), los importes iniciales y los aumentos de los elementos de PASIVO (Obligaciones);
el Capital, los Beneficios.

Vamos a representar gráficamente las normas expuestas:

D	Título	H
Importes iniciales Destinos de Activo Gastos Pérdidas Disminución de Obligaciones		Importes iniciales Obligaciones Aumentos de Obligaciones Salidas u orígenes de Activo Capital Beneficios

Teniendo en cuenta estos criterios, vamos a registrar en las cuentas los importes iniciales de un inventario de la familia en un momento del tiempo, en detalle y resumido:

ACTIVO

CAJA		120.000
Cabeza de familia	40.000	
Ama de la casa	80.000	
BANCOS		350.000
B. H. Mexicano	150.000	
Caja Postal	200.000	
MOBILIARIO		1.500.000
Equipo de sonido	250.000	
Televisión	90.000	
Vídeo	100.000	
Muebles	1.060.000	
INMUEBLES		6.500.000
Piso	2.300.000	
Chalé	4.200.000	
GASTOS DE CASA		80.000
Ropa	50.000	
Calzado	20.000	
Libros	10.000	
Total Activo		8.550.000

PASIVO		
ACREEDORES		80.000
Visa Postal	10.000	
El Corte Iglu	50.000	
Galerías Amigos	20.000	
PRESTAMOS		4.000.000
Caja Postal	3.500.000	
B. H. Mexicano	500.000	
HIPOTECAS		2.000.000
Total Pasivo		6.080.000

RESUMEN	
Total Activo	8.550.000
Total Pasivo	6.080.000
Capital Líquido	2.470.000

Veámoslo, resumido, el mismo inventario:

ACTIVO	
CAJA	120.000
BANCOS	350.000
MOBILIARIO	1.500.000
INMUEBLES	6.500.000
GASTOS DE CASA	80.000
Total Activo	8.550.000

PASIVO	
ACREEDORES	80.000
PRESTAMOS	4.000.000
HIPOTECAS	2.000.000
Total Pasivo	6.080.000

RESUMEN	
Total Activo	8.550.000
Total Pasivo	6.080.000
Capital Líquido	2.470.000

Vamos a hacer un inciso para definir las cuentas que sirven para recoger solamente el movimiento de los elementos de Activo y Pasivo, sin incluir resultados.



CUENTAS INTEGRALES (O DE VALORES)

Son aquéllas que sirven para registrar las variaciones que experimentan los distintos elementos del Patrimonio a los cuales controlan. Pueden ser de **ACTIVO** o de **PASIVO**, según que los elementos controlados sean bienes y derechos u obligaciones.

Empezaremos por los elementos del **ACTIVO**. A cada uno de ellos le abriremos una cuenta y colocaremos en el **DEBE** de la misma el importe que figura en el estadillo anterior:

D Cabeza de familia	H
40.000	

D Ama de Casa	H
80.000	

D B.H.M.	H
150.000	

D C. Postal	H
200.000	

D E. de sonido	H
250.000	

D Televisión	H
90.000	

D E. de vídeo	H
100.000	

D Piso	H
2.300.000	

D Chalé	H
4.200.000	

D Muebles	H
1.060.000	

D Libros	H
10.000	

D Calzado	H
20.000	

D Ropa	H
50.000	

Surge ahora, en la mayoría de los lectores que comienzan a estudiar la Contabilidad, una duda que se traduce en una sencilla pregunta y que quizá al lector también se le haya ocurrido:

¿Por qué todo lo que poseemos o tenemos lo ponemos en el Debe? Más concretamente: ¿por qué el dinero que poseemos, representado por la cuenta de Caja, se coloca en el Debe?

¿No sería más lógico anotarlo en el Haber, dejando el Debe para registrar lo que debemos?

Aparentemente este razonamiento tiene su lógica, pero si profundizamos en el significado de las cuentas veremos que es falso.

La cuenta de Caja no representa a la familia (sujeto Contable), sino que por el principio de personificación podemos suponer que representa al «Cajero» a quien hemos entregado para su custodia el dinero que poseemos, por lo que dicho Cajero es responsable del mismo y debe ese dinero a quien se lo ha entregado.

En la práctica el sujeto contable (familia, por ejemplo) puede ser a la vez cajero, pero a efectos contables se consideran separadas dichas funciones.

Si nos fijamos, por ejemplo, en la cuenta de Bancos, la cual representa y controla el dinero que tenemos depositado en la entidad de crédito, aquí se ve claro que el Banco debe al sujeto contable el dinero que éste le ha depositado.

Las restantes cuentas representan y controlan a sus respectivos elementos patrimoniales, que pueden personificarse en sus correspondientes responsables. Por tanto, **DEBEN** sus importes a quienes les ha entregado estos bienes para su custodia y administración.

Otra duda surge ahora con su correspondiente pregunta: si las CUENTAS representan a los elementos patrimoniales y no al sujeto contable, ¿no figura entonces dicho sujeto contable en los apuntes contables?, ¿no tiene alguna cuenta que lo represente? La respuesta es **SÍ**. Tiene una cuenta que le representa, a la que llamaremos **CAPITAL** y en la cual colocaremos en su Haber todo lo que tiene, es decir, la suma de los bienes y derechos que hemos anotado en el Debe de sus respectivas cuentas, o sea, el total del ACTIVO.

D Capital	H
	8.550.000

Como podemos observar, los importes iniciales de los elementos de activo los hemos anotado en las dos partes contrarias en que se divide una cuenta.

Por un lado, en el Debe de sus correspondientes cuentas y, por otro, en el Haber de la cuenta que representa al sujeto contable, o sea, el **CAPITAL**.

Al principio de este capítulo hemos hablado del método de Partida Doble. Trataremos de definirlo a continuación.



PARTIDA DOBLE

Sistema de Contabilidad consistente en apuntar o registrar las operaciones efectuadas en el Debe de una o varias cuentas y en el Haber de otra u otras varias, teniendo que ser igual el total adeudado al total acreditado.

Por esta razón recibe también el nombre de sistema digráfico.



Principios de la Partida Doble

1. El importe que se adeuda en una o varias cuentas (destino) ha de ser igual al abonado en una u otras varias cuentas (origen), y viceversa.
2. No hay deudor sin acreedor o no hay destino sin origen. Por consiguiente, siempre que se adeuda a una cuenta, necesariamente hay que acreditar por el mismo valor otra cuenta.
3. Toda cuenta que recibe o todo lo que entra es deudor, y toda cuenta que entrega o todo lo que sale es acreedor.

Como se verá, en las anotaciones de los importes del Activo hemos aplicado los principios de la Partida Doble, principios que continuaremos aplicando en las anotaciones de los importes iniciales del Pasivo y las sucesivas operaciones.

Antes de seguir adelante queremos remachar dos conceptos que hemos utilizado anteriormente en este capítulo: Sujeto Contable y Principio de personificación.



Sujeto Contable

Llamado también a veces «sujeto económico», es la familia, el comerciante, la empresa, etc., dueño del negocio, actividad familiar, etc., de cuya contabilidad estamos hablando.

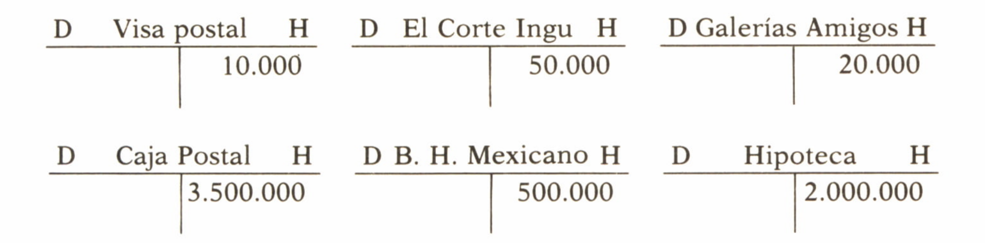


Principio de personificación

Es aquél en virtud del cual todas las cuentas se consideran personificadas y, por tanto, son capaces de recibir o entregar, aplicándoselas el tercer principio de la Partida Doble expresado anteriormente.

Una vez registrados los importes iniciales del Activo vamos a anotar en las cuentas los importes iniciales de todos y cada uno de los elementos del Pasivo.

Para ello, a cada elemento le abriremos su correspondiente cuenta y colocaremos en el Haber de la misma el importe que figura en el inventario:



D	Visa postal	H
		10.000

D	El Corte Ingu	H
		50.000

D	Galerías Amigos	H
		20.000

D	Caja Postal	H
		3.500.000

D	B. H. Mexicano	H
		500.000

D	Hipoteca	H
		2.000.000

Todas estas cuentas representan deudas u obligaciones de la familia **SUJETO CONTABLE**, por lo que, de acuerdo con los principios de la **PARTIDA DOBLE**, tenemos que adeudar el total de estos importes en la cuenta representativa de dicha familia (sujeto contable).

D	Capital	H
6.080.000		8.550.000

Por tanto, mientras en el Haber de la cuenta de CAPITAL se registra el total del Activo, en el Debe se anota el total del Pasivo.

Estas anotaciones pueden simplificarse anotando la diferencia o CAPITAL Líquido en el Haber de la cuenta de Capital, conforme indicamos anteriormente.

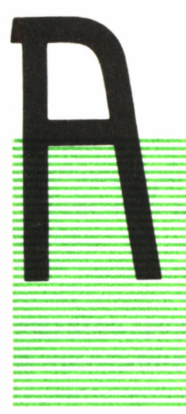
Esta es la forma que se utiliza normalmente en la realidad práctica. En este caso la cuenta de Capital quedaría así.

D	Capital	H
		2.470.000

Cumpliéndose también el principio de la Partida Doble, ya que el total adeudado en las cuentas de Pasivo y Capital.

¿Por qué se anotan en el Haber los importes iniciales de las cuentas de Pasivos del Patrimonio?, en nuestro préstamo Caja Postal, Galería Amigos, han de haber, han de recibir de la familia los importes que ésta les debe.

APUNTES EN LA CUENTA 4



CABAMOS de ver en el capítulo precedente el uso de las cuentas para registrar en las mismas cada uno de los importes iniciales. Vamos a dar un paso más y estudiar en este capítulo cómo se registran en las cuentas las operaciones sucesivas que realiza la familia en el transcurso de su vida cotidiana.

Para ello, lo primero que tenemos que hacer es analizar los Hechos Contables para determinar la forma correcta de contabilizarlos, es decir, de anotarlos en las cuentas. Este examen recibe el nombre de Análisis Precontable, el cual consiste en el estudio del Hecho Contable a fin de determinar:

a) Las cuentas que han de intervenir en la anotación del hecho contable.
b) El lugar (Debe o Haber) en que han de figurar los importes correspondientes en dichas cuentas.
c) El total del importe adeudado (destino), que ha de ser igual al total del importe abonado (origen).

Veamos, pues, y analicemos los Hechos Contables que expusimos anteriormente para tratar de registrarlos ahora a través de las correspondientes cuentas.

Recuerda que el primer Hecho Contable consistía en la compra de un equipo de música por 120.000 pesetas, pagando con talón.

Siguiendo los criterios del Análisis Precontable que acabamos de exponer, lo primero que tenemos que determinar son las cuentas que han de intervenir en la contabilización de este hecho contable.

Parece claro que si se compra un equipo de música (120.000 pesetas) y se paga con talón, hay un movimiento de dinero; por tanto, las cuentas que deben registrar dicho movimiento son:

Mobiliario y Bancos

¿En qué lugar de las mismas, Debe o Haber, han de anotarse los importes correspondientes?

Razonemos de la manera siguiente: si compramos un mueble (equipo de música), el mobiliario aumentará, habrá un «destino» y sabemos, por los criterios para el registro de operaciones en las cuentas expuestos en lo que antecede, que los aumentos o destinos de los elementos de ACTIVO se anotan en el DEBE.

Por tanto, si el mobiliario aumenta se contabilizará en el Debe. Por otra parte, si nosotros pagamos con talón, se produce una disminución de dinero «salida u origen» en la cuenta de Bancos, y sabemos que las disminuciones, salidas u orígenes de los elementos de ACTIVO se anotan en el Haber. Por tanto, si el Banco disminuye se anotará en el Haber.

Este hecho contable quedará registrado en las cuentas de la siguiente forma:



Obsérvese que en las cuentas aparecerán registradas, además de la operación de que se trate, las anotaciones que ya estuvieran anteriormente contabilizadas.

Como conclusión de este ejemplo, podemos ver que se cumplen los principios de la Partida Doble, expuestos anteriormente.

Si nosotros ahora cancelamos parte de la deuda de Galerías Amigos, mediante dinero que tenemos depositado en el Banco (por ejemplo, 10.000 pesetas), según el razonamiento hecho en el caso anterior, es evidente que, si sale dinero del Banco Hispano Mexicano y lo reciben las Galerías Amigos, las cuentas que funcionarán en este caso serán:

Banco Hispano Mexicano y Galerías Amigos

Del Banco sale el dinero, luego disminuye el importe de lo que allí tenemos depositado y, como sabemos, las disminuciones, salidas u orígenes de los elementos de Activo se anotan en el Haber de su cuenta respectiva. Tendremos, pues:

Si Banco disminuye, se contabilizará en el HABER.

«Galerías Amigos» también disminuye por cancelación de parte de nuestra deuda, pero como es una cuenta de Pasivo, esta disminución se anotará en el Debe.

Recordemos, pues, a este repecto el tercer principio de la partida doble:

«Toda cuenta que recibe o todo lo que "entra" es deudor».

Y las «Galerías Amigos», representada por su cuenta, recibe el dinero procedente del Banco. Por consiguiente:

Si Galerías Amigos disminuye o recibe, se contabilizará en el DEBE.

Como la cantidad a anotar tanto en el Debe como en el Haber es 20.000 pesetas, efectuaremos la siguiente contabilización en las cuentas mencionadas.

D	Banco Hispano Mexicano	H		D	Galerías Amigos	H
150.000		10.000	←——→	10.000		20.000

Hasta este momento, nos hemos dedicado exclusivamente a dar una breve exposición de qué es la Contabilidad, para qué sirve y cómo se van introduciendo los distintos hechos contables en los registros de cada cuenta, así como una representación gráfica de los mismos.

Todo esto no tendría sentido si no tratásemos de llevarlo a la práctica mediante un juego de programas sencillos que lo controlen o lo registren y nos permitan, cuando nosotros deseemos y de manera fácil, listarlo de ciertas formas. Para ello hemos tratado de elegir un lenguaje sencillo como el BASIC y el PASCAL, lenguajes que resultan muy introducidos en todos los ordenadores y que permiten al lector poco a poco ir confeccionando, mediante tecleo en su ordenador personal, un conjunto de programas que le permitirán realizar lo que pretende este libro, controlar su economía doméstica con el ordenador.

Establezcamos a continuación unos principios básicos, a saber:

— La cuenta de donde SALE el dinero (a la que, para evitar errores, se le dará un código de cuenta) y que llamaremos cuenta acreedora o cuenta origen, y a partir de este momento nos referiremos a ella con el nombre de ORIGEN.

— La cuenta a donde LLEGA el dinero, que llamaremos cuenta deudora o cuenta destino, y a partir de este momento nos referiremos a ella con el nombre de DESTINO.

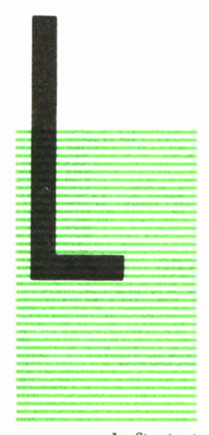
— La RAZON por la que efectuaremos este movimiento, es decir, el hecho contable que produce este movimiento y que a partir de ahora nos referiremos a él con el nombre de CONCEPTO.

— Por último, este movimiento se refleja mediante la cantidad en dinero que se relaciona con ambas cuentas y que a partir de ahora nos referiremos a ella con el nombre de PESETAS.

INTRODUCCION AL MODELO INFORMATICO 5



APLICACION CONTABLE



LA aplicación contable se basa en el establecimiento de una serie de Cuentas, que iremos agrupando en tablas y que podremos ampliar según nos haga falta. No obstante, para simplificar, al iniciarlas debemos prever el tamaño con una cierta amplitud.

Lógicamente, estas cuentas tienen un movimiento monetario entre ellas, que habrá que definir, y como quiera que cada persona que las utilice podría tener una forma distinta de nombrar la cuenta, para evitar problemas cada una se definirá con un código.

El código consta de uno a tres caracteres, con los que se harán todas las referencias a la cuenta que representa, y de una descripción literal que sustituirá al código (para mayor claridad) en los distintos informes de salida.

La aplicación está constituida por una serie de programas que manejan todos los datos de los diferentes hechos contables, datos que se registran en un fichero que contiene toda la información contable codificada.

En ese fichero se escriben los registros y de él se leen, para posteriormente producir diferentes tipos de informes.

Los movimientos de las cuentas (Hechos Contables) se definen especificando una serie de datos, como fecha, cuenta de origen, cuenta de destino y pesetas.

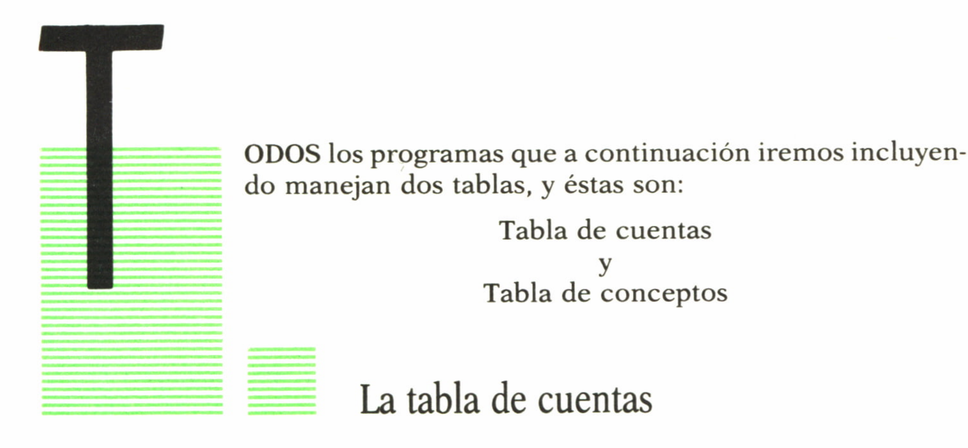
El fichero que contiene todos los movimientos contables deberá estar diseñado de la forma siguiente:

Nombre del Fichero	Apuntes. Mov.
Tipo de Fichero	Lectura/escritura

Tipo de Acceso	Lectura secuencial Escritura secuencial Adición secuencial por la cola.
Registros	Cada registro deberá contener la definición de un movimiento contable de acuerdo con el siguiente formato:

CAMPO	DESCRIPCION
FECHA	Número formado por (ddmmaa , siendo dd = día mm = mes aa = año
ORIGEN	Tres caracteres que representan el código de la cuenta acreedora o cuenta de «origen».
DESTINO	Tres caracteres que representan el código de la cuenta deudora o cuenta de «destino».
CONCEPTO	Tres caracteres que representan el código correspondiente al concepto por el que se produce el movimiento.
PESETAS	Cantidad numérica, que recoge el importe del movimiento.

PROGRAMA DE CARGA DE TABLAS 6



TODOS los programas que a continuación iremos incluyendo manejan dos tablas, y éstas son:

Tabla de cuentas
y
Tabla de conceptos

La tabla de cuentas

La tabla de cuentas se crea para poder referirnos a una cuenta mediante un código sencillo de tres caracteres, y con él manejar la aplicación de nuestra contabilidad casera.

En el programa escrito en Pascal, esta tabla estará recogida dentro de un fichero que llamaremos «Tabla. cta.» y que leeremos en forma de una matriz de 20 filas y tres columnas, donde cada cuenta se referencia por el índice de la fila que le corresponde.

Esta tabla obliga a crear otra tabla de descripción de los nombres de las cuentas, que utilizaremos siempre que efectuemos una salida o resultado en nuestra aplicación. Por ello, esta tabla tendrá el mismo número de filas que la tabla de cuentas, y por convenio vamos a establecer un ancho de 15 posiciones. Igualmente la incluiremos en un fichero que llamaremos «tabla. des.».

A continuación se presenta un ejemplo del procedimiento que podría utilizarse para crear las dos tablas mencionadas:

```
Repeat
 Repeat
```

```
sw := 0 ;
Repeat
 writeln('   ') ;
 writeln('Entre el codigo de cuenta....( 3 posiciones ) ' );
 readln(cta);
 if length(cta) < 3 then
  writeln(' Error. La longitud debe ser de tres letras. ' );
Until length(cta) = 3  ;
For I:=1 to 3 do
begin
 if not(cta[I] in['A'..'Z']) then sw := sw + 1 ;
end ;
If sw = 0 then
 mayus := true
 else
 writeln(' Por favor teclee letras Mayusculas ');
Until mayus = true ;
II := (II+1) ;
if cta = 'END' Then sicod := true ;
if sicod = false then
 begin
  tabctareg.tabcta[II] := cta;
  writeln(' ');
  writeln('Entre la descripción de la cuenta..',cta,' (max. 15 posiciones ) ');
  Readln(codd) ;
  for cc := length(codd)+1 to 15 do
   codd := codd+' ' ;
  tabdesreg.tabdes[II] := codd ;
  writeln(' ');
 end;
Until sicod = true ;
```

El procedimiento ejemplo de carga de códigos de cuentas y descripciones es repetitivo en su petición por pantalla; para ello comprueba:

En el código de cuenta

— Que la longitud del código de cuenta sea de tres caracteres.
— Que los caracteres tecleados sean letras mayúsculas.

En la descripción de cuenta

— Que la longitud debe ser de 15 caracteres (por ello, deberemos de teclear la descripción de la cuenta sin preocuparnos de su longitud, ya que el programa completa a 15 con blancos y si nos pasamos de 15, lo trunca a esa longitud).

Este procedimiento termina cuando, al leer el código de cuenta, detecta la clave «END».

En el caso del lenguaje Basic, el código de cuenta se encontrará en la variable «C$(20)», y en la variable «D$(20)» incluiremos la descripción de las mismas, por lo que su contenido se podría definir con un bucle que contenga una instrucción como:

```
READ C$(i),D$(i)
```

Que lea de un bloque de datos preparado previamente, como en el siguiente ejemplo:

```
DATA "BEC","BANESTO"
DATA "HAC","HACIENDA"
DATA "GG","GASTOS GENERALES"
ETC.
```



LA TABLA DE CONCEPTOS

La tabla de conceptos se crea para poder referirnos a un concepto mediante un código sencillo de tres caracteres, y con él manejar la aplicación de nuestra contabilidad casera.

En el programa escrito en Pascal, esta tabla estará recogida dentro de un fichero que llamaremos «Tabla. con.» y que leeremos en forma de una matriz (de 20 filas y tres columnas), donde cada cuenta se referencia por el índice de la fila que le corresponde.

Esta tabla obliga a crear otra tabla de descripción de los nombres de conceptos, que utilizaremos siempre que efectuemos una salida o resultado en nuestra aplicación. Esta segunda tabla tendrá el mismo número de filas que la tabla de conceptos, y por convenio vamos a establecer un ancho de 15 posiciones. Igualmente la incluiremos en un fichero que llamaremos «tabla. des.».

A continuación se presenta un ejemplo del procedimiento que podría utilizarse para crear las dos tablas mencionadas:

```
Procedure CODCTO ;
var
 sicod : boolean ;
Begin
 Clrscr;
 sicod := false ;
 III := 0 ;
 Repeat
  writeln(' ');
  writeln('Entre el codigo de concepto....(max. 3 posiciones ) ' );
  readln(cta) ;
  III := (III+1) ;
  if cta = 'END' then sicod := true ;
  if sicod = false then
   begin
    tabcptoreg.tabcpto[III] := cta ;
     writeln(' ');
     writeln('Entre la descripción del concepto..',cta,' (max. 15 posiciones) ');
     readln(codd) ;
     for cc := length(codd)+1 to 15 do
      codd := codd+' ' ;
     tabdpcptoreg.tabdpcpto[III] := codd;
     writeln(' ');
    end;
   Until sicod = true ;
  End ;
```

El procedimiento ejemplo de carga de códigos de conceptos y descripciones sólo comprueba:

En el código de concepto

— Que no hemos tecleado «END», ya que esto indicaría el final de la carga; aquí los códigos pueden tener la longitud de uno a tres caracteres.

En la descripción del concepto

— Que la longitud debe ser de 15 caracteres (por ello, deberemos de teclear la descripción de la cuenta sin preocuparnos de su longitud, ya que el programa completa a 15 con blancos y si nos pasamos de 15, lo trunca a esa longitud).

En el caso del lenguaje Basic, el código de concepto se encontrará en la variable «E$(20)» y en la variable «F$(20)» incluiremos la descripción

de cada concepto. Su contenido se podría definir con un bucle que contenga una instrucción como:

```
READ E$(i),F$(i)
```

Que lea de un bloque de datos preparado previamente, como en el siguiente ejemplo:

```
DATA "GAS","GAS BUTANO"
DATA "ELE","ELECTRICIDAD"
DATA "TEL","TELEFONO"
```

En la descripción de la salida que sigue se ha supuesto que se dispone de una pantalla de 80 caracteres.

Para los casos en que la pantalla tenga menos de 80 caracteres, hay que ajustar los comienzos de los distintos campos, de forma que la ruptura de líneas coincida con la separación entre los campos.

De esta forma los listados por impresora podrán seguir teniendo el aspecto deseado de las 80 columnas.

Seguidamente tratemos de iniciar este control mediante el juego de programas a que nos hemos referido anteriormente. Para ello debemos pensar que, cuando vamos a empezar a utilizar estos programas por primera vez, es preciso definir la situación inicial en que nos encontramos.

Así, pues, empezaremos por un programa de carga de tablas que podría programarse en PASCAL, y éste sería el siguiente:



PROGRAMA TABLAS, EN PASCAL

```
Program tablas(input,output);
type
  codcta    = string[3];
  coddes    = string[15];
  regcpto   = record
  tabcpto   : array[1..20] of codcta;
       end ;
  regdpcpto = record
```

```
   tabdpcpto : array[1..20] of coddes;
        end ;
   regcta    = record
   tabcta    : array[1..20] of codcta;
        end ;
   regdes    = record
   tabdes    : array[1..20] of coddes;
        end ;

 var
   Tablacta        : file of regcta;
   tabctareg       : regcta;
   Tablades        : file of regdes;
   tabdesreg       : regdes;
   Tablacpto       : file of regcpto;
   tabcptoreg      : regcpto;
   tabladpcpto     : file of regdpcpto;
   tabdpcptoreg    : regdpcpto;
   cta             : string[3];
   cod             : string[1];
   cc,II,I,III     : Integer;
   codd            : string[15] ;

   Procedure CIERRE ;
     Begin
        close (tablacpto);
        close (tabladpcpto);
        close(tablacta) ;
        close(tablades) ;
      End ;
   Procedure INICIALIZA ;
    Begin
     With tabctareg,tabdesreg,tabcptoreg,tabdpcptoreg Do
      for I := 1 to 20 Do
       begin
        tabcta[I] := '   ';
        tabdes[I] := '               ';
        tabcpto[I]:= '   ';
        tabdpcpto[I]:= '               ';
       end;
    End ;
    Procedure CODICTA ;
    var
     sicod : boolean ;
     mayus : boolean ;
     sw    : integer ;
    Begin
     clrscr ;
```

```
writeln('Deme los codigos cta y descripción ') ;
writeln(' y teclee END para salir del bucle..');
writeln(' ');
 sicod := false ;
 mayus := false ;
 II := 0 ;
Repeat
 Repeat
  sw := 0 ;
  Repeat
   writeln('   ') ;
   writeln('Entre el codigo de cuenta....( 3 posiciones ) ' );
   readln(cta);
   if length(cta) < 3 then
    writeln(' Error. La longitud debe ser de tres letras. ' );
  Until length(cta) = 3  ;
  For I:=1 to 3 do
  begin
   if not(cta[I] in['A'..'Z']) then sw := sw + 1 ;
  end ;
  If sw = 0 then
   mayus := true
   else
   writeln(' Por favor teclee letras Mayusculas ');
  Until mayus = true ;
 II := (II+1) ;
 if cta = 'END' Then sicod := true ;
 if sicod = false then
  begin
   tabctareg.tabcta[II] := cta;
   writeln(' ');
   writeln('Entre la descripción de la cuenta..',cta,' (max. 15 posiciones ) ');
   Readln(codd) ;
   for cc := length(codd)+1 to 15 do
    codd := codd+' ' ;
   tabdesreg.tabdes[II] := codd ;
   writeln(' ');
  end;
 Until sicod = true ;
End ;

Procedure CODCTO ;
var
 sicod : boolean ;
Begin
 Clrscr;
 sicod := false ;
 III := 0 ;
 Repeat
```

```
writeln(' ');
writeln('Entre el codigo de concepto....(max. 3 posiciones ) ' );
readln(cta) ;

III := (III+1) ;
if cta = 'END' then sicod := true ;
if sicod = false then
 begin
  tabcptoreg.tabcpto[III] := cta ;
   writeln(' ');
   writeln('Entre la descripción del concepto..',cta,' (max. 15 posiciones) ');
   readln(codd) ;
   for cc := length(codd)+1 to 15 do
    codd := codd+' ' ;
   tabdpcptoreg.tabdpcpto[III] := codd;
   writeln(' ');
  end;
 Until sicod = true ;
End ;

Procedure LISTACTA ;
 Begin
  clrscr;
  WITH tabctareg,tabdesreg Do
   Begin
    writeln('Tabla de cuentas y sus Descripciones');
    writeln('------------------------------------');
    writeln('   ') ;
    writeln(' ');
    Writeln('Codigo Cuenta    Descripción    ');
    writeln('-------------    -----------    ');
    writeln(' ');
    For I:= 1 to II do
     writeln('    ',tabcta[I],'              ',tabdes[I]);
    writeln(' ');
   writeln(' ¿ Esta de acuerdo con la tabla ?..(s/n) ');
   readln(cod);
   If cod = 's' then
    begin
     write(tablacta,tabctareg);
     write(tablades,tabdesreg);
     close(tablacta) ;
     close(tablades) ;
    end
     else
      begin
       writeln(' no se salva la tabla de cuentas ');
       writeln(' entrela de nuevo ......');
```

```
          close(tabldes);
          close(tablacta);
         end;
      end ;
    End;

  Procedure LISTACTO ;
   Begin
    Clrscr ;
    writeln('Tabla de conceptos y Descripciones');
    writeln('----------------------------------');
    writeln('   ');
    writeln('   ');
    writeln('Codigo Concepto     Descripción  ');
    writeln('---------------     ------------ ');
    writeln('   ');
    With tabcptoreg,tabdpcptoreg Do
     Begin
       For I:= 1 to III do
        writeln('    ',tabcpto[I],'             ',Tabdpcpto[I]);
       writeln('   ');
       writeln('   ');
       writeln('¿ Esta Ok la tabla ? ( s/n )   ');
       Readln(cod);
       if cod = 's' then
        begin
         write(tablacpto,tabcptoreg);
         write(tabladpcpto,tabdpcptoreg);
         close (tablacpto);
         close (tabladpcpto);
        end
         else
          begin
           writeln(' no se salva la tabla de conceptos ');
           writeln(' entrela de nuevo ......');
           close (tablacpto);
           close (tabladpcpto);
          end ;
     End ;
   End ;


  {********* Cuerpo del Programa ***********}

BEGIN

  assign(tablacta,'tabla.cta');
  rewrite(tablacta);
```

```
    assign(tablades,'tabla.des');
    rewrite(tablades);
    assign(tablacpto,'tabla.cto');
    rewrite(tablacpto);
    assign(tabladpcpto,'tabla.dto');
    rewrite(tabladpcpto);
    INICIALIZA ;
    CODICTA ;
    CODCTO ;
    LISTACTA ;
    LISTACTO ;
    CIERRE ;
  END.
```



Explicación del Programa Tablas: Carga de los códigos y conceptos

El programa comprende al principio declaraciones de variables y tipos, de estructuras o formatos internos de variables, que el programa va a utilizar. Algunas de estas declaraciones son estructuras en forma de tabla, otras definen la composición del registro de los ficheros que contienen las tablas que hemos visto anteriormente:

Tablacta, Tablacpto, Tablades y Tabladcpto.

En el cuerpo del programa empieza el programa con la instrucción «BEGIN». Seguidamente se empieza por asignar nombres de ficheros y abrirlos para escritura mediante las instrucciones siguientes:

assign(tablacta,'tabla.cta'); rewrite(tablacta);	{ asignación apertura para escribir
assign(tablades,'tabla.des'); rewrite(tablades);	{ asignación apertura para escribir
assign(tablacpto,'tabla.cto'); rewrite(tablacpto);	{ asignación apertura para escribir
assign(tabladpcpto,'tabla.dto'); rewrite(tabladpcpto);	{ asignación apertura para escribir

En la asignación, y dependiendo de nuestro ordenador, podemos también indicar en qué disco queremos grabar el fichero.



Procedimiento Inicializa

Este sirve para ir limpiando las diferentes tablas de caracteres raros de la memoria, poniéndolo todo a blancos.

La instrucción:

With tabctareg,tabdesreg,tabcptoreg,tabdpcptoreg Do

se utiliza para poder referirnos posteriormente a la tabla que deseemos, de la forma sencilla siguiente:

writeln(' ',tabcta[I],' ',tabdes[I]);

si no, tendríamos que escribir la misma instrucción de la forma siguiente:

writeln(' ',tabctareg.tabcta¿Iñ,' ',tabdesreg.tabdes¿Iñ);

La instrucción CLRSCR se utiliza para limpiar la pantalla. Esta instrucción puede variar según la máquina y el compilador de Pascal que estemos utilizando. En caso de duda, se recomienda al lector que consulte el Manual correspondiente.



Procedimiento Listacta

Se usa para proceder al listado por pantalla de las tablas de cuentas y descripción de las mismas.

El bloque de sentencias que sigue a continuación se utiliza para poder comprobar por pantalla si lo tecleado es correcto o no, pero solamente hasta el valor de la variable «II», que es la que controla el número de códigos incluidos en procedimiento CODICTA.

```
WITH tabctareg,tabdesreg Do
 Begin
  writeln('Tabla de cuentas y sus Descripciones');
  writeln('-----------------------------------');
  writeln('  ') ;
  writeln('  ');
  Writeln('Codigo Cuenta    Descripción   ');
  writeln('-------------    -----------   ');
  writeln('  ');
  For I:= 1 to II do
  begin
   writeln('    ',tabcta[I],'              ',tabdes[I]);
  end ;
End ;
```

El bloque de sentencias que sigue a continuación se utiliza para poder dar la conformidad a la tabla, en cuyo caso se graba en el fichero, se cierra y continuamos con el programa. Por el contrario, si no damos la conformidad, no se graba, pero también se cierra el fichero y se continúa, porque en este caso la tabla no está bien.

```
writeln(' ¿ Esta de acuerdo con la tabla ?..(s/n) ');
readln(cod);
If cod = 's' then
 begin
  write(tablacta,tabctareg);
  write(tablades,tabdesreg);
  close(tablacta) ;
  close(tablades) ;
 end
  else
   begin
    writeln(' no se salva la tabla de cuentas ');
    writeln(' entrela de nuevo ......');
    close(tablades);
    close(tablacta);
   end;
```

Procedimiento Listacto

Se usa para proceder al listado por pantalla de las tablas de conceptos y descripción de los mismos.

El bloque de sentencias que sigue a continuación se utiliza para poder visualizar la otra tabla, que, al igual que en el caso anterior, sólo se verá en la pantalla, hasta el valor de la variable «III».

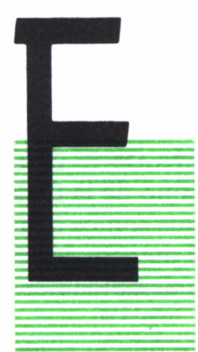
```
writeln('Tabla de conceptos y Descripciones');
writeln('----------------------------------');
writeln('  ');
writeln('  ');
writeln('Codigo Concepto     Descripción  ');
writeln('---------------     ------------ ');
writeln('  ');
With tabcptoreg,tabdpcptoreg Do
 Begin
   For I:= 1 to III do
   begin
    writeln('    ',tabcpto[I],'              ',Tabdpcpto[I]);
   end ;
 end ;
```

En el siguiente grupo de instrucciones tratamos solamente de dar nuestra conformidad a la segunda tabla, para poder cerrar el fichero y terminar:

```
With tabcptoreg,tabdpcptoreg Do
 Begin

   writeln('   ');
   writeln('¿ Esta Ok la tabla ? ( s/n )   ');
   Readln(cod);
   if cod = 's' then
    begin
     write(tablacpto,tabcptoreg);
     write(tabladpcpto,tabdpcptoreg);
     close (tablacpto);
     close (tabladpcpto);
    end
     else
      begin
       writeln(' no se salva la tabla de conceptos ');
       writeln(' entrela de nuevo ......');
       close (tablacpto);
       close (tabladpcpto);
      end ;
 end ;
```

PROGRAMA INICIAR FICHERO 7



ESTE programa se arrancará cuando demos comienzo a nuestra aplicación de contabilidad, lógicamente si nos hemos leído anteriormente, de la teoría contable, nos habremos dado cuenta de que antes de iniciar nada, habrá que establecer un inventario casero, que refleje de alguna manera nuestra situación en ese momento, para irla posteriormente actualizándola.

Este programa será el que creará el fichero que controla todos los movimientos, el cual le hemos dado el nombre de **APUNTES. MOV**.

El programa debe registrar en el fichero recién creado un registro por cada una de las distintas cuentas, con el saldo que hayamos establecido a las mismas, y que de alguna manera tendrá un sentido real y coherente.

Esta información del saldo de las cuentas se introduce en ellas mediante un registro de movimiento ficticio «**SAL**» (saldo inicial, contemplado para este tipo de situaciones), que contiene en el código de la cuenta tanto si es una cuenta de tipo **ORIGEN** (acreedora) o **DESTINO** (deudora); por tanto, intervendran:

campo «**FECHA**»
campo «**ORG**»
campo «**DES**»
campo «**CON**»
campo «**PTAS**»

Por supuesto, antes de ejecutar el inicio en sí de las cuentas, para establecer los saldos de arranque, habrá primero que ir cargando todas y cada una de las distintas tablas que hemos ido indicando en lo que se refiere a cuentas, descripción de las mismas, conceptos y sus descripciones. Para generar la información de arranque de las cuentas el programa de Carga (iniciar) deberá pedir la fecha de comienzo (que no tiene por qué coinci-

dir con la fecha del día); después, el programa deberá pedir para cada uno los códigos de cuentas en la tabla cargada:

Saldo inicial para la cuenta... «BEC».

Una vez tecleado el importe inicial de la cuenta y verificado que lo tecleado es número, pasaremos, pues, a identificar si el arranque del saldo inicial es:

DEUDOR (destino) o ACREEDOR (origen).

Pues como sabemos, una cuenta tiene la posibilidad de, según el hecho contable originado, tener uno u otro tipo de movimiento.

Por ejemplo, iniciar el arranque de BANESTO: el importe existente es de 60.000 pesetas, el programa deberá registrar estas 60.000 pesetas como saldo inicial deudor (destino), ya que, según estudiamos anteriormente, por el concepto de personificación de cuentas el dinero en BANCOS es deudor nuestro, si cojiéramos e iniciáramos la cuenta de Préstamo de la Caja Postal, el saldo inicial sería acreedor (origen), ya que nosotros somos deudores de la Caja Postal, luego ella es un acreedor nuestro, el manejo de estos conceptos no debe de confundirnos, si nos hemos leído detenidamente el capítulo de cuentas.

Y por último, la cantidad de dinero de dicho movimiento que lo almacenaremos en el campo PESETAS; con todos estos datos tenemos conformado el contenido del registro, que llevaremos al fichero de Apuntes. Mov., donde secuencialmente se almacenan todos los diferentes movimientos.



PROGRAMA INICIAR, EN PASCAL

```
Program Iniciar(input);
 Type
      regcta = record
       tabcta : array[1..20] of string[3] ;
         end ;

VAR
    origen    : string[3] ;
    destino   : string[3] ;
    concepto : string[3] ;
    ptas     : real ;
    pesetas  : string[10] ;
    dd : string[2] ;
    mm : string[2] ;
    aa : string[2] ;
```

```
 num : boolean ;
tablacta   : File of regcta ;
  tabctareg : regcta ;
 Apuntes    : Text ;
cta,cod  : string[3] ;
chk  : string[1] ;
I,c,cc : integer ;
dia,mes,year : Integer ;
fechae : string[6] ;

 Procedure Abrir ;
 Begin
  assign(tablacta,'tabla.cta') ;
  reset(tablacta);
  assign  (apuntes,'apuntes.mov') ;
  rewrite (apuntes) ;


    clrscr ;
    read(tablacta,tabctareg) ;
    writeln('   ')  ;
    writeln(' Visión de la tabla de cuentas leida ') ;
    writeln('  ') ;
    For I := 1 to 10 do
     begin
      writeln(tabctareg.tabcta[I]) ;
     end ;
    writeln('   ') ;
    close(tablacta) ;
   End ;

  Procedure DIARIO ;
  Var
   dia : integer ;
   sw  : boolean ;
  Begin
    sw := false ;
    Repeat
      writeln('   ') ;
      writeln('Deme el dia...') ;
      writeln('  ') ;
      read(dia) ;
      writeln('  ') ;

    if (not(dia in[1..31])) then
      begin
       writeln('  ' ) ;
       writeln('no valido el dia....',dia:2) ;
```

```
          sw := false ;
         end
         else
          sw := true ;
      Until sw = true ;
 str(dia:2,dd) ;
 if dd[1] = ' ' then dd[1] := '0' ;
End ;

Procedure MENSUAL ;
Var
 mes : integer ;
 sw  : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   ') ;
    writeln('Deme el mes...') ;
    writeln('  ') ;
    read(mes) ;
    writeln('  ') ;
   if (not(mes in[1..12])) then
     begin
      writeln('  ' ) ;
      writeln(' no valido el mes ....',mes:2) ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
 str(mes:2,mm) ;
 if mm[1] = ' ' then mm[1] := '0' ;
End ;

Procedure ANUAL ;
Var
 anno : integer ;
 sw   : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   ') ;
    writeln('Deme el año...') ;
    writeln('  ') ;
    read(anno) ;
    writeln('  ') ;
   if (not(anno in[80..99])) then
     begin
```

```
     writeln('  ' ) ;
     writeln(' no valido el año ....',anno:2) ;
     sw := false ;
    end
    else
     sw := true ;
 Until sw = true ;
  str(anno:2,aa) ;
  fechae := concat(dd,mm,aa) ;
 End ;

  Procedure CUENTA ;
  var
  sw : boolean ;
  w,I  : integer ;
  Begin
   sw := false ;
   Repeat
   Repeat
    Repeat
    writeln('    ') ;
    writeln('Digame si es cuenta  Origen (org) o Destino (des) ');
    writeln('  ');
    read(cod) ;
    w := 0 ;
    for I:= 1 to 3 do
     begin
      if not(cod[I] in['o','r','g','d','e','s']) then w := w +1 ;
     end ;
    if w > 0 then
     begin
      writeln('    ') ;
      writeln('ERROR. cuenta... ', cod:3,' mal, tecleelo de nuevo por favor  ') ;
      writeln('    ') ;
      w := 0 ;
     end
     else sw := true ;
   until sw = true ;
    if length(cod) < 3 then
     begin
      writeln('    ') ;
      writeln('ERROR. longitud mal. tecleelo de nuevo por favor ...',cod:3 ) ;
      writeln('    ') ;
     end ;
    Until length(cod) = 3 ;
    If (cod = 'org')  then
     begin
      origen  := cta ;
```

```
  destino :='   ';
  end
 else
  begin
   if (cod = 'des' ) then
   begin
    destino := cta ;
    origen  :='   ';
   end
  else
      begin
       writeln(' ') ;
       writeln(' ERROR. no ha tecleado Vd bien ') ;
       writeln(' debe ser (org) o (des)... tecleelo de nuevo ') ;
       writeln(' ') ;
       sw := false ;
      end ;
    end ;
   Until sw = true ;
 End ;

Procedure Identificador ;
 Begin
   num := false ;
   c:= 2 ;
   Repeat
   if not(pesetas[c] = ' ' ) then
    begin
     cc:= (c-1) ;
     pesetas[cc] := '.' ;
     num := true ;
    end
    else c := (c+1) ;
   Until num = true ;
  End ;

Procedure Proceso ;
  var
   fin : boolean ;
   sw  : boolean ;
Begin
 fin := false ;
 I := 1 ;
 While  (fin = false) do
  begin
   cta := tabctareg.tabcta[I] ;
   if not(cta = '   ') then
    begin
```

```
      writeln('   ') ;
      writeln('Saldo inicial para la cuenta...',cta:3 );
      writeln('   ');
      read(ptas) ;
      Cuenta ;
      str(ptas:10:0,pesetas) ;
      Identificador ;
      writeln(' ') ;
      writeln(' contenido del registro a grabar ......') ;
      writeln(' ') ;
      writeln(fechae,' ',origen,' ',destino,' ',concepto,' ',pesetas) ;
      writeln(Apuntes,fechae,' ',origen,' ',destino,' ',concepto,' ',pesetas) ;
      sw := false ;
      Repeat
       writeln('   ') ;
       writeln('  teclee (s) para seguir o (f) para acabar ');
       read(chk) ;
       If not(chk[1] in['s','f']) then
        writeln('ERROR. por favor intentelo de nuevo ')
       else
        sw := true ;
      Until sw = true ;
      if chk = 'f' then fin := true ;
      I := I+1 ;
     end
     else
      fin := true ;
   End ;
 End;

{********** cuerpo del programa **************}
BEGIN
  Abrir ;
  Diario ;
  Mensual;
  Anual;
  concepto  :='SAL';
  Repeat
   ptas := 0 ;
   origen   :='   ';
   destino  :='   ';
   Proceso ;
  Until chk = 'f' ;
  close(Apuntes);

End.
```



EXPLICACION DEL PROGRAMA INICIAR

El programa comprende al principio declaraciones de variables y tipos de estructuras o formatos internos de variables, que el programa va a utilizar. Algunas de estas declaraciones son estructuras en forma de tabla, otras definen la composición del registro de los ficheros que contienen las tablas.

Hemos establecido una serie de procedimientos para ir paso a paso avanzando en la inicialización del fichero de referencia.



Procedimiento Abrir

Comenzamos el procedimiento con la asignación de dos ficheros, el primero es la tabla de códigos de cuenta y el segundo es un fichero de texto donde almacenaremos los apuntes contables.

```
Procedure Abrir ;
Begin
 assign(tablacta,'tabla.cta') ;
 reset(tablacta);
 assign  (apuntes,'apuntes.mov') ;
 rewrite (apuntes) ;
```

Antes de iniciar el proceso efectuamos un listado, por pantalla de la tabla de cuentas, con objeto de recordar el nombre y orden en el que se incluyeron en el fichero de códigos de cuentas, y así proporcionar una ayuda al lector recordando los códigos de las cuentas; las instrucciones son las siguientes:

```
clrscr ;
read(tablacta,tabctareg) ;
writeln('   ') ;
writeln(' Visión de la tabla de cuentas leida ') ;
writeln('  ') ;
For I := 1 to 10 do
 begin
  writeln(tabctareg.tabcta[I]) ;
 end ;
writeln('   ') ;
close(tablacta) ;
```



Procedimiento Diario

Empieza pidiendo el día, comprueba que lo tecleado está dentro de los días del mes; en caso contrario, indica «día no válido», y vuelve a pedir el día; en caso de teclear bien el día, pasa al siguiente procedimiento; las instrucciones son las siguientes:

```
sw := false ;
Repeat
  writeln('   ') ;
  writeln('Deme el dia...') ;
  writeln('   ') ;
  read(dia) ;
  writeln('   ') ;
 if (not(dia in[1..31])) then
   begin
    writeln('   ' ) ;
    writeln('no valido el dia....',dia:2) ;
    writeln('   ') ;
    sw := false ;
   end
   else
    sw := true ;
Until sw = true ;
```

convertimos el día leído en carácter de dos posiciones y lo almacenamos en la variable «dd»:

str(día:2,dd) ;

si la primera posición primera de la variable «dd» es blanco, le incluimos un cero.

if dd[1] = " then dd[1] := '0' ;



Procedimiento Mensual

En este bloque sólo comprobamos que el mes tecleado sea correcto y válido, ya que, en caso contrario, nos lo indica y nos lo pide otra vez; las instrucciones son las siguientes:

```
sw := false ;
Repeat
  writeln('   ') ;
```

```
    writeln('Deme el mes...') ;
    writeln(' ') ;
    read(mes) ;
    writeln(' ') ;
   if (not(mes in[1..12])) then
     begin
      writeln(' ' ) ;
      writeln(' no valido el mes ....',mes:2) ;
      writeln(' ') ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
```

convertimos el mes leído en carácter de dos posiciones y lo almacenamos en la variable «mm»:

str(mes:2,mm) ;

si la primera posición primera de la variable «mm» es blanco, le incluimos un cero:

str(mes:2,mm) ;
if mm[1] = ' ' then mm[1] := '0' ;



Procedimiento Anual

Con este procedimiento comprobamos que el año tecleado sea correcto y esté comprendido entre los años 80 y 99, que así lo hemos establecido por convenio en el programa; en caso contrario, nos lo indica y nos vuelve a pedir el año; las instrucciones son las siguientes:

```
  sw := false ;
  Repeat
    writeln(' ') ;
    writeln('Deme el año...') ;
    writeln(' ') ;
    read(anno) ;
    writeln(' ') ;
   if (not(anno in[80..99])) then
     begin
```

```
                writeln('  ' ) ;
                writeln(' no valido el año ....',anno:2) ;
                writeln('  ') ;
                sw := false ;
              end
              else
                sw := true ;
            Until sw = true ;
```

convertimos el año leído en carácter de dos posiciones y lo almacenamos en la variable «aa»:

str(year:2,aa) ;

Con los procedimientos Diario, Mensual y Anual almacenamos la fecha en las variables «dd», «mm» y «aa», las cuales son unidas y asignadas a la variable «fechae», con la instrucción:

fechae := concat(dd,mm,aa) ;

que servirá para todo el resto del programa.

Inicializamos la variable «concepto» con el contenido de «SAL», que igualmente nos servirá para todo el programa.



Procedimiento Proceso

Este procedimiento utiliza la variable «fin», definida como booleana, para controlar el fin de la entrada de datos al fichero de Apuntes. mov., la cual se puede dar de dos maneras:

— Por vacío en la tabla de cuentas.

Comenzamos leyendo el primer código de la tabla de cuentas, que verificamos si es blanco, en cuyo caso el programa debe finalizar, pues indica que no hay más cuentas en la tabla.

Instrucciones:

```
fin := false ;
I := 1 ;
While (fin = false) do
 begin
   cta := tabctareg.tabcta[I] ;
```

```
                if not(cta = '   ') then
                 begin
                  writeln('   ') ;
                  ..................
                  ..................
                  ..................
                 end
                 else
                  fin := true ;
```

— Tecleando una «f» para acabar.

```
          sw := false ;
          Repeat
           writeln('   ') ;
           writeln('  teclee (s) para seguir o (f) para acabar ');
           read(chk) ;
           If not(chk[1] in['s','f']) then
            begin
             writeln('  ') ;
             writeln('ERROR. por favor intentelo de nuevo ') ;
             writeln( '  ');
            end
           else
            sw := true ;
          Until sw = true ;
          if chk = 'f' then fin := true ;
```

De acuerdo a la cuenta en la tabla, este procedimiento pide el saldo inicial para ella, según las siguientes instrucciones:

```
            cta := tabctareg.tabcta[I] ;
            if not(cta = '   ') then
             begin
              writeln('   ') ;
              writeln('Saldo inicial para la cuenta...',cta:3 );
              writeln('   ');
              read(ptas) ;
              .................
              .................
```

A continuación llama al procedimiento Cuenta, y posteriormente convierte la cifra leída de real a caracteres, llamando a continuación al procedimiento Identificador.

Veamos las instrucciones:

```
                    ....................
                    ....................
                    ....................
                    Cuenta ;
                    str(ptas:10:0,pesetas) ;
                    Identificador ;
                    writeln(' ') ;
                    ....................
                    ....................
```

Efectuamos un listado del registro que vamos a grabar para posteriormente grabarlo en el fichero Apuntes.

Veamos las instrucciones:

```
....................
....................
....................
writeln(' ') ;
writeln(' contenido del registro a grabar ......') ;
writeln(' ') ;
writeln(fechae,' ',origen,' ',destino,' ',concepto,' ',pesetas) ;
writeln(Apuntes,fechae,' ',origen,' ',destino,' ',concepto,' ',pesetas) ;
writeln(' ') ;
....................
....................
```



Procedimiento Identificador

Este procedimiento se utiliza para incluir en el registro un separador que nos permita identificar la posición, en dónde empieza el número en el fichero de Texto (Apuntes).

Para ello se ha definido la variable «núm», como (booleana), y que sólo puede adquirir los valores de «verdadero» o «falso» («true» y «false», en inglés).

Veamos a continuación sus instrucciones:

```
Procedure Identificador ;
 Begin
   num := false ;
```

```
c:= 2 ;
Repeat
if not(pesetas[c] = ' ' ) then
 begin
  cc:= (c-1) ;
  pesetas[cc] := '.' ;
  num := true ;
 end
 else c := (c+1) ;
Until num = true ;
End ;
```

La variable «ptas.» la definimos como real, para que almacene números mayores de 32767, aunque posteriormente la convertiremos en un literal de 10 posiciones para escribirlo en el fichero.



Procedimiento Cuenta

Después de leer el importe en pesetas, tenemos que indicar si es origen o destino; las instrucciones que a continuación se indican realizan dicha operación.

```
writeln('   ') ;
writeln('Digame si es cuenta  Origen (org) o Destino (des) ');
writeln('  ');
read(cod) ;
.....................
.....................
.....................
.....................
```

Seguidamente comprobamos el código de destino u origen; tecleado se compone de las letras «org» o «des», escritas en minúsculas, escribiendo un mensaje de error; cuando esté mal, todo esto se hace mediante:

```
w := 0 ;
for I:= 1 to 3 do
 begin
  if not(cod[I] in['o','r','g','d','e','s']) then w := w +1 ;
```

```
  end ;
 if w > 0 then
  begin
   writeln('   ') ;
   writeln('ERROR. cuenta... ', cod:3,' mal, tecleelo de nuevo por favor  ') ;
   writeln('   ') ;
   ....................
   ....................
```

A continuación verificamos si la longitud es de tres letras, y en el caso de no serlo, nos dará un mensaje de error; veamos las instrucciones:

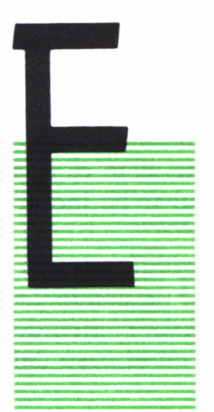
```
 Repeat
  if length(cod) < 3 then
   begin
    writeln('   ') ;
    writeln('ERROR. longitud mal. tecleelo de nuevo por favor ...',cod:3 ) ;
    writeln('   ') ;
   end ;
  Until length(cod) = 3 ;
```

Por último, si lo tecleado corresponde a las palabras «org» o «des», efectúa la asignación correspondiente de la cuenta, origen o destino, según corresponda. Veamos las instrucciones:

```
If (cod = 'org')  then
 begin
  origen  := cta ;
  destino :='   ';
 end
else
 begin
  if (cod = 'des' ) then
   begin
    destino := cta ;
    origen  :='   ';
   end
  else
   begin
    writeln(' ') ;
    writeln(' ERROR. no ha tecleado Vd bien ') ;
```

```
      writeln(' debe ser (org) o (des)... tecleelo de nuevo ') ;
      writeln(' ') ;
      sw := false ;
     end ;
  end;
```

PROGRAMA APUNTES AL FICHERO 8



EMPEZAREMOS por la definición de los apuntes contables, teniendo en cuenta, como es natural, el principio de hecho contable que se ha originado para saber si su imputación a una cuenta es deudor (destino), o sea, entra o acreedor (origen) o sea, sale de la cuenta.

Este programa deberá añadir al fichero que contiene los movimientos iniciales, por la cola los nuevos registros; para ello el programa realizará una rutina (bucle) en el que deberá ir pidiendo al usuario los datos necesarios para identificar el movimiento, es decir, cuando hablábamos de la estructura del registro del fichero indicábamos los campos de que se componía el registro; por tanto, ésta descripción la debe pedir el programa.

En la pantalla el usuario va definiendo el movimiento que quiere registrar, y para ello le van apareciendo sucesivamente las siguientes preguntas:

Deme el día
Deme el mes
Deme el año
Dígame el código de la cuenta Origen
Dígame el código de la cuenta Destino
Deme el importe del movimiento

movimientos y que se da por finalizada la entrada de datos. Entonces se cierra el fichero y termina el programa.

Si el número tecleado en el campo día es distinto de cero, el programa comprueba si el día tecleado está dentro del período 1 a 31. Si no es así, entonces aparecerá en pantalla el mensaje siguiente:

día inválido (y se repite el mensaje de petición de día).

Cuando se ha tecleado un día que es correcto y que está dentro del intervalo (1 a 31), el programa pasa a continuación con la petición del siguiente campo (MES).

El usuario tecleará el mes, y el programa comprueba si está comprendido entre 1 y 12 (repitiendo la petición de mes si no es correcto).

Posteriormente el programa pide el año y él comprueba igualmente si la respuesta está comprendida entre 80 y 99, que es el intervalo que hemos establecido para este programa.

A continuación el programa deberá pedir el código de origen, que constará de tres letras. Si tecleamos un código que no se encuentra en la tabla de códigos, como puede suceder que el usuario no se acuerde del código que tecleó cuando lo incluyó en la tabla, el programa podría ofrecer una ayuda al usuario, limpiando la pantalla y presentando la tabla de cuentas y la descripción de las mismas, con lo que el usuario tendría un medio rápido de poder recordar cómo es el código que debe teclear. Esto es una sugerencia para posibles mejoras de este programa, pero no se ha incluido en el mismo.

Igualmente ocurrirá con el código de destino, que el programa también comprueba con objeto de ayudar al usuario cuando esté cargando datos. (Recordemos que el código de las diferentes cuentas está contenido en la variable C$, para el caso del Basic, y en un fichero, en el caso del Pascal.)

Pasamos, pues, a pedir el nuevo campo, o sea, el concepto, que dijimos anteriormente iba a estar codificado, para evitar los errores correspondientes. Por tanto, igual que en el caso de los campos de Origen y Destino, el programa también comprueba si el código tecleado es o no correcto (recordemos que los códigos de concepto se hallan contenidos en la variable E$, para el caso del Basic, y en un fichero para el Pascal).

Por último, el programa pide la cantidad en pesetas para incluirla en el campo del mismo nombre y comprueba que lo que se teclea es un número positivo sin decimales.

Después que el programa ha realizado todas estas comprobaciones, deberá añadir el nuevo registro por la cola del fichero y seguidamente continúa con el bucle de seguir pidiendo más apuntes, hasta que se defina un cero en el campo día, que, como ya se ha dicho, sirve para que el programa lo detecte como final de la entrada de datos, cierre el fichero y dé por finalizada la sesión.

Veamos el ejemplo en Pascal:

Programa Apuntes: Añadir registros al Fichero de apuntes.mov

```
Program Apuntes(input);
const
```

```
  Name = 'Apuntes.mov';
Type
  FileNameType = string[66];
  codcta = string[3] ;
regcta = record
tabcta : array[1..20] of codcta ;
   end ;
regcon = record
tabcon : array[1..20] of codcta ;
   end ;
Var
  Tablacta : File of regcta ;
  Tablacon : File of regcon ;
  tabctareg : regcta ;
  tabconreg : regcon ;
  Old,New: text;
  line   : string[30] ;
  nombre : string[7] ;
  con    : string[3] ;
  org    : string[3] ;
  des    : string[3] ;
  cta    : string[3] ;
  cod    : string[1] ;
  I      : Integer ;
  ptas   : real ;
  pesetas : string[10] ;
  fechae : string[6] ;
  dd    : string[2] ;
  mm    : string[2] ;
  aa    : string[2] ;
  num : boolean ;
  mas : boolean ;
{-------------------------------------------------------------}
 Procedure Cuenta ;
 var
  sicta  : boolean ;
 Begin
  Repeat
   repeat
    writeln('   ') ;
    writeln('Digame el codigo de la cuenta...',nombre) ;
    writeln('   ') ;
    readln(cta) ;
    if length(cta) < 3 then
     begin
      writeln('   ') ;
      writeln('ERROR, codigo ',cta,' deben ser 3 caracteres ') ;
      writeln('intentelo de nuevo. Por favor' ) ;
```

```
       end ;
      until length(cta) = 3 ;
      sicta := false ;
      For I := 1 to 20 do
       begin
       if tabctareg.tabcta[I] = cta then
        begin
         if nombre = 'ORIGEN ' then
          org := cta ;
         if nombre = 'DESTINO' then
          des := cta ;
         sicta := true ;
        end ;
       end ;
       if sicta = false then
          begin
           writeln('   ') ;
           writeln('ERROR. codigo',cta,' no está en la tabla ') ;
           writeln('compruebelo y tecleelo de nuevo') ;
           writeln('    ');
          end ;
      Until sicta = true ;
   End ;
  {--------------------------------------------------------------}
  Procedure Concepto ;
  Var
   sicon : boolean ;
   Begin
    sicon := false ;
   Repeat
    writeln('   ') ;
    writeln('Deme el codigo de concepto') ;
    Readln(cta) ;
    writeln('   ') ;
    sicon := false ;
    For I := 1 to 20 do
     begin
      if tabconreg.tabcon[I] = cta then
       begin
        con := cta ;
        sicon := true ;
       end ;
     end ;
      if sicon = false then
       begin
        writeln('   ') ;
        writeln('ERROR. codigo',cta,' no está en la tabla ') ;
        writeln('compruebelo y tecleelo de nuevo') ;
```

```
        writeln('    ');
       end ;
   Until sicon = true ;
   if length(con) < 3  then con := concat(con,' ') ;
  End ;
{-------------------------------------------------------------}
   Procedure Importe ;
    Begin
     writeln('   ') ;
     writeln('Deme el importe del movimiento') ;
     writeln('   ') ;
     Readln(ptas) ;
     str(ptas:10:0,Pesetas) ;
    End ;
{-------------------------------------------------------------}
   Procedure Identificador ;
   var
    c,cc : integer ;
    Begin
      num := false ;
      c:= 2 ;
      Repeat
      if not(pesetas[c] = ' ' ) then
       begin
        cc:= (c-1) ;
        pesetas[cc] := '.' ;
        num := true ;
       end
       else c := (c+1) ;
      Until num = true ;
     End ;
{-------------------------------------------------------------}
  Function Exist (Fichero: FileNameType): boolean;
   var F: file of byte;
  begin
   assign(F,Fichero);
   {$I-} Reset (F) {$I+};
   Exist:= (IOresult=0);
   close (F)
  end;
{-------------------------------------------------------------}
  Procedure Abrir_copiar ;
   Begin
  assign(tablacta,'tabla.cta'); reset(tablacta) ;
  assign(tablacon,'tabla.cto'); reset(tablacon) ;
  read(tablacta,tabctareg) ;
  read(tablacon,tabconreg) ;
  close(tablacta) ; close(tablacon) ;
```

```
assign(NEW,'apts_new.mov');
rewrite(NEW) ;
if Exist (Name) then            {* Si ya existía, copiarlo al nuevo *}
   begin
    assign (Old,Name);
    reset (Old);
    while not Eof(Old) do
    begin
     readln (Old,Line);
     writeln (New,Line)
    end;
    close (Old)
   end;
End ;
{-----------------------------------------------------------------}
 Procedure DIARIO ;
 Var
  dia : integer ;
  sw  : boolean ;
 Begin
   sw := false ;
   Repeat
     writeln('   ') ;
     writeln('Deme el dia...') ;
     writeln('  ') ;
     read(dia) ;
     writeln('  ') ;
    if (not(dia in[1..31])) then
      begin
       writeln('  ' ) ;
       writeln('no valido el dia....',dia:2) ;
       sw := false ;
      end
      else
       sw := true ;
   Until sw = true ;
  str(dia:2,dd) ;
  if dd[1] = ' ' then dd[1] := '0' ;
 End ;
{-----------------------------------------------------------------}
 Procedure MENSUAL ;
 Var
  mes : integer ;
  sw  : boolean ;
 Begin
   sw := false ;
   Repeat
     writeln('   ') ;
```

```
     writeln('Deme el mes...') ;
     writeln('  ') ;
     read(mes) ;
     writeln('  ') ;
    if (not(mes in[1..12])) then
      begin
       writeln('  ' ) ;
       writeln(' no valido el mes ....',mes:2) ;
       sw := false ;
      end
      else
       sw := true ;
   Until sw = true ;
  str(mes:2,mm) ;
  if mm[1] = ' ' then mm[1] := '0' ;
 End ;
{-----------------------------------------------------------------}
Procedure ANUAL ;
Var
 anno : integer ;
 sw   : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   ') ;
    writeln('Deme el año...') ;
    writeln('  ') ;
    read(anno) ;
    writeln('  ') ;
   if (not(anno in[80..99])) then
     begin
      writeln('  ' ) ;
      writeln(' no valido el año ....',anno:2) ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
 str(anno:2,aa) ;
 fechae := concat(dd,mm,aa) ;
End ;
{*********** Cuerpo del Programa **************}
 BEGIN
  Abrir_copiar ;
  mas := true ;
 Repeat
  org := '   ';
  des := '   ';
```

```
        Diario ;
        Mensual ;
        Anual ;
        nombre := 'ORIGEN ' ;
        Cuenta ;
        nombre := 'DESTINO' ;
        Cuenta ;
        Concepto ;
        Importe ;
        Identificador ;
        writeln(New,Fechae,' ',org,' ',des,' ',con,' ',Pesetas) ;
        writeln(' ') ;
        writeln('¿ Quiere registrar mas apuntes (s/n) ?') ;
        writeln('  ') ;
        Readln(cod);
        If cod='n' then mas := false ;
       Until mas = false ;

          close(NEW) ;
          erase (OLD) ;
          rename (NEW,Name) ;
     END.
```



EXPLICACION DEL PROGRAMA

El programa comprende al principio declaraciones de variables y tipos, de estructuras o formatos internos de variables, que el programa va a utilizar. Algunas de estas declaraciones son estructuras en forma de tabla, otras definen la composición del registro de los ficheros que contienen las tablas.



Procedimiento Abrir-copiar

Hemos establecido una serie de procedimientos y función para ir primero verificando si el fichero Apuntes.mov existe, en cuyo caso lo asignaremos y abriremos como «OLD» y mientras no sea final de fichero, iremos copiando registro a registro en otro «NEW», que abriremos con el nom-

bre provisional de Apts_new.mov; esto lo realizamos con las instrucciones siguientes:

```
assign(NEW,'apts_new.mov');
rewrite(NEW) ;
if Exist (Name) then               {* ver Función Exist *}
   begin
    assign (Old,Name);
    reset (Old);
    while not Eof(Old) do
    begin
     readln (Old,Line);
     writeln (New,Line)
    end;
    close (Old)
   end;

   La función que a continuación se lista sirve para verificar, si
   existe el fichero a copiar o no.

Function Exist (Fichero: FileNameType): boolean;
 var F: file of byte;
  begin
   assign(F,Fichero);
   {$I-} Reset (F) {$I+};
   Exist:= (IOresult=0);
   close (F)
  end;
```

Los procedimientos Identificador, Diario, Mensual y Anual funcionan igual que en el caso del programa anterior (Iniciar).



Procedimiento Cuenta

Este procedimiento pide el código de la cuenta, empezando primero por el de la cuenta «ORIGEN», y después la cuenta «DESTINO», asignando en cada caso a las variables «org» o «des» el código correspondiente; naturalmente, como siempre, se comprueba si la longitud de lo tecleado

es de tres caracteres, o si está o no en la tabla de códigos; veamos las instrucciones:

Longitud:

```
nombre := 'ORIGEN ' ;
 repeat
  writeln('   ') ;
  writeln('Digame el codigo de la cuenta...',nombre) ;
  writeln('   ') ;
  readln(cta) ;
  if length(cta) < 3 then
   begin
    writeln('   ') ;
    writeln('ERROR, codigo ',cta,' deben ser 3 caracteres ') ;
    writeln('intentelo de nuevo. Por favor' ) ;
   end ;
  until length(cta) = 3 ;
```

Tabla:

```
Repeat
  sicta := false ;
  For I := 1 to 20 do
   begin
   if tabctareg.tabcta[I] = cta then
    begin
     if nombre = 'ORIGEN ' then
      org := cta ;
     if nombre = 'DESTINO' then
      des := cta ;
     sicta := true ;
    end ;
   end ;
   if sicta = false then
      begin
       writeln('   ') ;
       writeln('ERROR. codigo',cta,' no está en la tabla ') ;
       writeln('compruebelo y tecleelo de nuevo') ;
       writeln('    ');
      end ;
 Until sicta = true ;
```



Procedimiento Concepto

Este procedimiento pide el concepto o naturaleza del movimiento, el cual es comprobado si se encuentra en la tabla correspondiente, o lo pide de nuevo caso de no estar también completa con un blanco en el supuesto de ser menor de tres caracteres; veamos a continuación las instrucciones:

Tabla:

```
sicon := false ;
For I := 1 to 20 do
 begin
  if tabconreg.tabcon[I] = cta then
   begin
    con := cta ;
    sicon := true ;
   end ;
 end ;
```

No en la Tabla

```
if sicon = false then
 begin
  writeln('   ') ;
  writeln('ERROR. codigo',cta,' no está en la tabla ') ;
  writeln('compruebelo y tecleelo de nuevo') ;
  writeln('    ');
 end ;
```

Una vez comprobado el concepto, pasamos a pedir el importe del movimiento con las siguientes instrucciones:

```
writeln('   ') ;
writeln('Deme el importe del movimiento') ;
writeln('   ') ;
Readln(ptas) ;
```

Con la siguiente instrucción convertimos el número real en un string:

str(ptas:10:0,Pesetas) ;



Grabación en Fichero

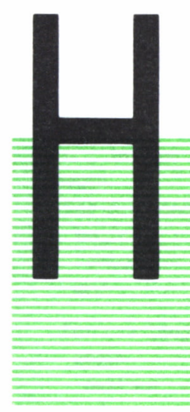
Escribo en el «NEW» el registro, y pregunto si se quiere algún otro apunte; si contestamos con una «s», vuelvo a empezar; en caso contrario, cierro el fichero «NEW», borro el fichero «OLD» y renombro el fichero «NEW», y termina el programa.

```
 mas := true ;
Repeat
................
................
................

 writeln(New,Fechae,' ',org,' ',des,' ',con,' ',Pesetas) ;
 writeln(' ') ;
 writeln('¿ Quiere registrar mas apuntes (s/n) ?') ;
 writeln(' ') ;
 Readln(cod);
 If cod='n' then mas := false ;
Until mas = false ;

   close(NEW) ;
   erase (OLD) ;
   rename (NEW,Name) ;
```

PROGRAMA LISTA UNA CUENTA 9



HASTA este momento nos hemos limitado a la carga de tablas de cuentas, mediante unos códigos que las identifican; hemos cargado también tablas de conceptos igualmente codificadas, y posteriormente nos hemos metido con el arranque inicial de las referidas cuentas, para luego llegar a la anotación de registros, mediante la petición de los datos de los apuntes o hechos contables que diariamente se van produciendo, guardándolo todo en fichero y tabla.

Por ello, viene ahora cómo poder obtener la información que deseemos consultando el fichero.

Este programa sirve para obtener del fichero de movimientos (Apuntes) especificado por el usuario una lista de los movimientos correspondientes a una determinada cuenta, incluyendo una columna adicional con el nuevo saldo que provoca cada nuevo movimiento.

Ejemplo del resultado de este programa:

Movimiento de la cuenta: BANESTO.

Fichero APUNTES.MOV:

Mov.	**Fecha**	**Origen**	**Destino**	**Concepto**	**Ptas.**	**Saldo**
25	01/09/86		BANESTO	saldo inicial	200.600	
59	23/09/86	BANESTO	Inmobilia.	letra	50.600	150.000
123	25/09/86	BANESTO	CAJA	talón	25.000	125.000
145	27/09/86	HACIENDA	BANESTO	transferencia	77.890	202.890

Expliquemos los movimientos:

El primero indica que el 1-09-86 se abrió la cuenta con un saldo inicial de 200.600 pesetas; aparece en blanco la columna Origen debido a que refleja que Banesto nos debe ese importe o, dicho de otra manera, hemos depositado (destinado) en Banesto ese importe.

El segundo indica que Banesto ha originado (ha salido de la cuenta) el importe cuyo destino ha sido la cuenta de Inmobiliaria y cuyo concepto ha sido el pago de la letra.

El tercero indica que Banesto ha originado, mediante el concepto de talón, la salida del dinero indicado con destino a la caja.

El cuarto indica o refleja que Banesto ha recibido (destino) una transferencia de dinero procedente de la cuenta de Hacienda.

Para hacer este pequeño informe, el programa tiene que hacer una búsqueda por todo el fichero, para poder obtener de la cuenta en cuestión todos los movimientos que se hayan producido o en los que haya intervenido.

```
Program cuenta(output) ;
 Type
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;
 regdes = record
  tabdes : array[1..20] of coddes ;
     end ;
 regcon = record
  tabcon : array[1..20] of codcta ;
     end ;
 regdcon = record
  tabdcon : array[1..20] of coddes ;
     end ;

Var
  Tablacta   : file of regcta ;
  Tablades   : file of regdes ;
  Tablacon   : file of regcon ;
  Tabladcon  : file of regdcon ;
  tabctareg   : regcta ;
  tabdesreg   : regdes ;
  tabconreg   : regcon ;
  tabdconreg  : regdcon ;
  Apuntes     : TEXT ;
  Listar      : Text ;
  sicta,mascta : boolean ;
  org : string[3] ;
  des : string[3] ;
  con : string[3] ;
  nombre : string[6] ;
  sumat :  string[10];
  cantidad : string[10];
  cta   :  string[3]  ;
```

```
  lista :  string[82] ;
  origen : string[15] ;
  destino : string[15] ;
  con1 : string[3] ;
  concepto :string[15] ;
  suma,II : real ;
  I,mov,c,chk,ic,emp,id : Integer ;
  cifra : string[10] ;
  dd : string[2] ;
  mm : string[2] ;
  aa : string[2] ;
  a : string[1] ;

Procedure ANCHO ;
 VAR
  c : integer ;
 begin
  For c:= length(origen)+1 to 15 do
   begin
    origen := concat( origen,' ');
   end ;

  For c:= length(destino)+1 to 15 do
   begin
    destino := concat(destino,' ') ;
   end ;
  For c:= length(concepto)+1 to 15 do
   begin
    concepto := concat(concepto,' ') ;
   end
 end ;

Procedure BUSCA ;
 VAR
   cc : Integer ;
 begin
  For cc := 1 to 20 do
   begin
    if org = tabctareg.tabcta[cc]  then
         origen := tabdesreg.tabdes[cc] ;
   end ;
  For cc := 1 to 20 do
   begin
    if des = tabctareg.tabcta[cc] then
     destino := tabdesreg.tabdes[cc] ;
   end ;
  For cc := 1 to 20 do
   begin
    if con = tabconreg.tabcon[cc] then
```

```
      concepto := tabdconreg.tabdcon[cc] ;
    end
  end ;

Procedure CIERRE;
    begin
     close (Apuntes) ; close (Listar) ;
    end ;

Procedure Nosal ;
  begin
    if (org = cta ) then
      begin
       if nombre = 'pasivo' then suma := suma+II ;
       if nombre = 'activo' then suma := suma-II ;
       BUSCA ;
       ANCHO ;
      end ;
     if (des = cta ) then
       begin
        if nombre = 'pasivo' then suma := suma-II ;
        if nombre = 'activo' then suma := suma+II ;
        BUSCA ;
        ANCHO ;
       end ;
      str(suma:10:0,sumat) ;
      writeln(Listar,mov:2,' ',dd:2,'/',mm:2,'/',aa:2,' ',origen,' ',destino,' ',
     concepto,cifra:9,sumat:9);
     end ;

 Procedure Empezar ;
  Begin
    if (org = cta) or (des = cta) then
      begin
       if emp = 1 then nosal ;
       if (con = 'SAL' ) then
        begin
         nombre := 'pasivo' ;
         if des = cta then nombre := 'activo' ;
         BUSCA ;
         ANCHO ;
  writeln(listar,mov:2,' ',dd:2,'/',mm:2,'/',aa:2,' ',origen,' ',destino,' ','
  Saldo inicial ',' ',cifra:9,' ',cifra:9 );
      suma :=( suma+II ) ;
      emp :=1 ;
     end ;
    end
   End ;
```

```
Procedure LEER ;
 VAR
    cc : integer ;
    line : string[35] ;
 Begin
    Readln(Apuntes,line) ;
    mov := (mov+1) ;
    cantidad := copy(line,20,10) ;
    cc := pos('.',cantidad);
    cifra := copy(line,(20+cc),(10-cc));
    val(cifra,II,chk) ;
    if (not (chk = 0 ) ) then
      begin
       writeln('ERROR numero al convertir ');
       CIERRE
       end ;
    org := copy(line,8,3) ;
    des := copy(line,12,3) ;
    con1 := copy(line,16,3) ;
    dd := copy(line,1,2) ;
    mm := copy(line,3,2) ;
    aa := copy(line,5,2) ;
    con := con1 ;
     if con1[3] = ' ' then
      begin
        con := copy(con1,1,2);
        con := concat(con,' ') ;
      end ;
 End ;

Procedure PROCESO ;

Begin
 assign(Apuntes,'apuntes.mov') ; reset(Apuntes) ;
 assign(Listar,'listar.cta')   ; rewrite(Listar) ;
 while not(eof(apuntes)) do
   begin
    con1 := '   ' ;
    LEER ;
    EMPEZAR ;
   end
End ;

Procedure Listado ;
 Begin
  assign (Listar,'Listar.cta') ; reset (Listar);
  clrscr ;
  writeln(' Listado para la cuenta ....',tabdesreg.tabdes[id] ) ;
  writeln( ' ') ;
```

```
    writeln('  ');
    writeln('NM FECHA     ORIGEN          DESTINO        CONCEPTO       PESETAS
    writeln('-- -------- --------------- -------------- -------------  ---------
    ACUMULADO ');
    ----------');

   writeln('  ');

   WHILE not EOF(Listar) Do
     Begin
      readln(Listar,lista) ;
      writeln(lista) ;

     end ;
   close (Listar) ;

  End ;

     { **** Cuerpo del Programa **** }

 BEGIN
  assign(tablacta,'tabla.cta') ; reset(tablacta) ;
  assign(tablades,'tabla.des') ; reset(tablades) ;
  assign(tablacon,'tabla.cto') ; reset(tablacon) ;
  assign(tabladcon,'tabla.dto') ; reset(tabladcon) ;

  read(tablacta,tabctareg) ; read(tablades,tabdesreg) ;
  read(tablacon,tabconreg) ; read(tabladcon,tabdconreg) ;
  close (tablacta) ; close (tablacon) ;
  close (tablades) ; close (tabladcon) ;
  mascta := true ;
  Repeat
   mov := 0 ; suma := 0 ; emp := 0 ;  sicta := false ;
   clrscr ;
   REPEAT
    Repeat
     writeln('   ') ;
     writeln('Deme la cuenta a buscar ') ;
     writeln('   ') ;
     read(cta) ;
     writeln('   ') ;
      if length(cta) < 3 then
      writeln('ERROR. en la longitud, teclee de nuevo la cuenta ') ;
    Until length(cta) = 3 ;

    For I := 1 to 20 do
     begin
      If tabctareg.tabcta[I] = cta then
       begin
```

```
     sicta := trûe ;
     id := I ;
    end ;
  end ;

     if sicta = false then writeln('Error. ',cta:3,' no encontrada en tabla ');
     UNTIL sicta = true ;
    PROCESO ;
    CIERRE ;
    if sicta = true then LISTADO ;
    Writeln('  ') ;
    writeln('  ') ;
    writeln(' ¿ Quiere alguna otra cuenta  ? ( s / n ) ') ;
    readln(a) ;
     if a = 'n' then mascta := false ;
   Until mascta = false ;

  END.
```



EXPLICACION DEL PROGRAMA

Este programa consta de ocho procedimientos, cada uno de los cuales realiza una determinada misión, repetitiva cada vez.



Procedimiento ANCHO

Este procedimiento sólo pretende completar con blancos, hasta 15 en total, el contenido de las variables (origen, destino y concepto), con las instrucciones siguientes:

```
For c:= length(origen)+1 to 15 do
 begin
  origen := concat( origen,' ');
 end ;
For c:= length(destino)+1 to 15 do
 begin
  destino := concat(destino,' ') ;
 end ;
For c:= length(concepto)+1 to 15 do
 begin
  concepto := concat(concepto,' ') ;
 end
```



Procedimiento BUSCA

Su misión es localizar, dentro de las tablas de cuentas y conceptos, su pareja con lo leído en el fichero; veamos alguna instrucción:

```
For cc := 1 to 20 do
 begin
  if org = tabctareg.tabcta[cc]  then
       origen := tabdesreg.tabdes[cc] ;
 end ;
For cc := 1 to 20 do
 begin
  if des = tabctareg.tabcta[cc] then
   destino := tabdesreg.tabdes[cc] ;
 end ;
For cc := 1 to 20 do
 begin
  if con = tabconreg.tabcon[cc] then
   concepto := tabdconreg.tabdcon[cc] ;
 end
```



Procedimiento CIERRE

Cierra los ficheros (Apuntes) y (Listar).



Procedimiento LEER

Este procedimiento es muy importante, ya que es el que efectúa la lectura del fichero y copia los datos a las diferentes variables. Veamos a continuación algunas instrucciones con detalle:

Primero leemos un registro y lo almacenamos en «line».

Readln(Apuntes,line) ;

Copiamos de la variable «line», a partir de la posición 20, diez caracteres, que almacenamos en la variable «cantidad».

cantidad := copy(line,20,10) ;

El bloque que viene a continuación es para poder detectar el separador incluido en el fichero, con objeto de que al convertir el número con-

tenido en la variable «cantidad» de caracteres a número real, no nos dé error, y todo sean números.

```
cc := pos('.',cantidad);
cifra := copy(line,(20+cc),(10-cc));
val(cifra,II,chk) ;
  if (not (chk = 0 ) ) then
    begin
      writeln('ERROR numero al convertir ');
      CIERRE
    end ;
```

Las otras instrucciones de este procedimiento sirven tan sólo para trasladar a las diferentes variables los datos del registro, a partir de sus posiciones correspondientes.



Procedimiento EMPEZAR

Con este procedimiento iniciamos la comprobación de si alguna de las dos cuentas del registro leído es igual a la cuenta que tratamos. También verificamos si es saldo inicial o no, para poner en la variable «nombre» el indicativo de si es una cuenta de «activo o de pasivo», ya que, dependiendo de esto, las demás partidas que intervengan en el saldo acumulado se sumarán o restarán, según corresponda.

Después llamamos al procedimiento **BUSCA**, **ANCHO** y a continuación grabamos en el fichero listar el registro, efectuamos la suma en el acumulador, ponemos el indicador «**EMP**» con valor uno, para detectarlo en la próxima pasada. Por tanto, si el registro no fuese un saldo inicial, al detectar el indicador a uno, llamará al procedimiento NOSAL.



Procedimiento NOSAL

Con este procedimiento tratamos todos los registros que no sean saldo inicial, y dependiendo de cómo esté activada la variable «nombre», sumaremos o restaremos del acumulador (suma). Llevaremos el último resultado a la variable (sumat), en formato de string, y grabaremos el registro formado.



Procedimiento PROCESO

Este tan sólo asigna el fichero Apuntes, lo abre, además asigna un fichero provisional, llamado Listar, que abre en modo escritura y que es el que nos servirá luego para listar el resultado.

Leeremos todo el fichero Apuntes, secuencialmente, e iremos llamando por cada lectura a los procedimientos LEER y EMPEZAR, hasta que se detecte el fin de fichero, lo cual se efectúa con:

While not EOF(Apuntes) Do



Procedimiento LISTADO

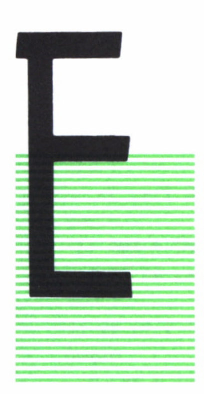
Con este procedimiento lo que hacemos es, después de cerrado el fichero Apuntes y Listar, volvemos a abrir el fichero Listar, en modo de lectura y comenzamos a leer un registro e imprimirlo, hasta que se detecte el fin de fichero:

While not EOF(Listar) Do

Todos estos procedimientos están incluidos dentro del cuerpo del programa, en donde lo primero que hacemos es pedir la cuenta a buscar, chequeando la longitud de lo tecleado y comprobando si se encuentra en la tabla de cuentas.

También, después de obtenido el resultado, damos opción de pedir más cuentas. Esto lo controlamos con la variable «mascta», que la hemos definido como booleana, terminando el bucle cuando tenga el valor de «false».

PROGRAMA LISTA DE MOVIMIENTOS 10



ESTE programa sirve para obtener del fichero de movimientos (Apuntes) especificando por el usuario una lista de movimientos correspondientes a un número de ellos, empezando en uno y a partir de ahí todos los que se quieran, o se detecte final de fichero.

Ejemplo de resultado de este programa:

Listado para los movimientos 1 al 4.

Mov.	Fecha	Origen	Destino	Concepto	Ptas.
1	10/10/86		BANESTO	saldo inicial	125.000
2	10/10/86		CAJA CASA	saldo inicial	80.000
3	10/10/86		C. AHORROS	saldo inicial	200.000
4	10/10/86		HISPANO	saldo inicial	50.000



PROGRAMA ESCRITO, EN PASCAL

```
Program listamvtos(output) ;
 Type
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;
 regdes = record
  tabdes : array[1..20] of coddes ;
     end ;
```

```
  regcon = record
   tabcon : array[1..20] of codcta ;
      end ;
  regdcon = record
   tabdcon : array[1..20] of coddes ;
      end ;

Var
   Tablacta    : file of regcta ;
   Tablades    : file of regdes ;
   Tablacon    : file of regcon ;
   Tabladcon   : file of regdcon ;
   tabctareg    : regcta ;
   tabdesreg    : regdes ;
   tabconreg    : regcon ;
   tabdconreg   : regdcon ;
   Apuntes     : TEXT ;
   Listar      : Text ;
   simov,primero : boolean ;
   org : string[3] ;
   des : string[3] ;
   con : string[3] ;
   cantidad : string[10];
   cta    :  string[3]  ;
   lista :  string[80] ;
   origen : string[15] ;
   destino : string[15] ;
   con1 : string[3] ;
   concepto :string[15] ;
   II : real ;
   move,movf,I,movi,nmov,c,chk,nummov : Integer ;
   cifra : string[10] ;
   line : string[35] ;
   dd : string[2];
   mm : string[2] ;
   aa : string[2] ;

Procedure ANCHO ;
  VAR
   c : integer ;
  begin
   For c:= length(origen)+1 to 15 do
    begin
     origen := concat( origen,' ');
    end ;

   For c:= length(destino)+1 to 15 do
    begin
     destino := concat(destino,' ') ;
```

```
    end ;
   For c:= length(concepto)+1 to 15 do
    begin
     concepto := concat(concepto,' ') ;
    end
  end ;

Procedure BUSCA ;
  VAR
    cc : Integer ;
  begin
   For cc := 1 to 20 do
    begin
     if org = tabctareg.tabcta[cc]  then
          origen := tabdesreg.tabdes[cc] ;
    end ;
   For cc := 1 to 20 do
    begin
     if des = tabctareg.tabcta[cc] then
      destino := tabdesreg.tabdes[cc] ;
    end ;
   For cc := 1 to 20 do
    begin
     if con = tabconreg.tabcon[cc] then
      concepto := tabdconreg.tabdcon[cc] ;
    end
  end ;

Procedure CIERRE;
    begin
     close (Apuntes) ; close (Listar) ;
    end ;

Procedure Empezar ;
  Begin
   BUSCA ;
   ANCHO ;
   writeln(Listar,move:3,'  ',dd:2,'/',mm:2,'/',aa:2,'   ',origen,' ',destino,' ',
   concepto,cifra:10) ;
  end ;

  Procedure COPIA ;
   VAR
    cc : integer ;
   Begin
     cantidad := copy(line,20,10) ;
     cc := pos('.',cantidad);
     cifra := copy(line,(20+cc),(10-cc));
     val(cifra,II,chk) ;
```

```
      if (not (chk = 0 ) ) then
        begin
         writeln('ERROR numero al convertir ');
         CIERRE
        end ;
      org := copy(line,8,3) ;
      des := copy(line,12,3) ;
      con1 := copy(line,16,3) ;
      dd := copy(line,1,2) ;
      mm := copy(line,3,2) ;
      aa := copy(line,5,2) ;
      con := con1 ;
       if con1[3] = ' ' then
        begin
         con := copy(con1,1,2);
         con := concat(con,' ') ;
        end ;
  End ;
Procedure POSICION ;
 Begin
  nmov := nmov +1 ;
  Readln(Apuntes,line) ;
  if movi = nmov then
   begin
    COPIA ;
    simov := true ;
   end ;
  if primero = true then
   begin
    if nmov <= movf then
       begin
        COPIA ;
        move :=move +1 ;
        simov := true ;
       end
   end
 End ;

Procedure PROCESO ;

Begin
 assign(apuntes,'apuntes.mov') ; reset(apuntes) ;
 clrscr ;
 writeln('  ') ;
 writeln('Deme el numero de movimiento inicial .. ') ;
 writeln('  ' );
 readln(movi) ;
 move := movi ;
```

```
    writeln('¿Cuantos movimientos quiere apartir del movimiento ',movi:3 );
    writeln('  ');
    readln(nummov) ;
    movf := (movi + nummov) ;
    while not(eof(apuntes))  do
      begin
       POSICION ;

     if simov = true then
      begin
       EMPEZAR ;
       primero := true ;
       simov   := false ;
      end ;
    end
  End ;

 Procedure Listado ;
  Begin
   assign (Listar,'Listar.cta') ; reset (Listar);
   clrscr ;
   writeln(' Listado para los movimientos numeros ',movi:3,' a ',(movf   :3  ) ;
   writeln( '  ') ;
   writeln('  ');
   writeln('MOV  FECHA      ORIGEN          DESTINO         CONCEPTO
   writeln('---  --------   --------------- --------------- ---------------
  PESETAS  ');
  ---------- ');
   writeln('  ');

    WHILE not EOF(Listar) Do
      Begin
       readln(Listar,lista) ;
       writeln(lista) ;

      end ;
    close (Listar) ;

   End ;

     { **** Cuerpo del Programa **** }

 BEGIN
  assign(tablacta,'tabla.cta') ; reset(tablacta) ;
  assign(tablades,'tabla.des') ; reset(tablades) ;
  assign(tablacon,'tabla.cto') ; reset(tablacon) ;
  assign(tabladcon,'tabla.dto') ; reset(tabladcon) ;
  assign(Listar,'Listar.cta') ; rewrite(Listar) ;
```

```
    nmov := 0 ; primero := false ; simov := false ;
    read(tablacta,tabctareg) ; read(tablades,tabdesreg) ;
    read(tablacon,tabconreg) ; read(tabladcon,tabdconreg) ;
    close (tablacta) ; close (tablacon) ;
    close (tablades) ; close (tabladcon) ;
      PROCESO ;
      Cierre  ;
      LISTADO ;
   END.
```



EXPLICACION DEL PROGRAMA

Este programa consta de ocho procedimientos, cada uno de los cuales realiza una determinada misión.

Los procedimientos: ANCHO, BUSCA, CIERRE y LISTADO funcionan igual que en el caso anterior.



Procedimiento Proceso

Asigna y abre el fichero Apuntes, pide después el número del movimiento desde el cual se quiere listar y el número de ellos, esto se almacena en las varables «movi» y «movf»; a partir de aquí empezamos llamando al procedimiento **POSICION** y **COPIA**.

Activamos el procedimiento **EMPEZAR** cuando la variable «simov» tiene valor de verdadero (en inglés, «true»).



Procedimiento Posición

Controla, cada vez que lee del fichero, si es un movimiento que está dentro del intervalo pedido; caso de serlo, efectúa la llamada al procedimiento **COPIA**, poniendo en la variable «simov» el valor de verdadero (en inglés, «true»).



Procedimiento Copia

Copia los datos del registro seleccionando las variables correspondientes.

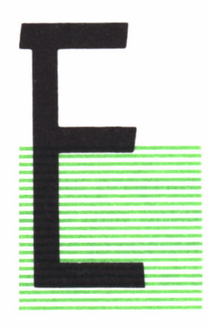


Procedimiento Empezar

Prepara el registro seleccionado y lo graba.

Como podrá observarse, el cuerpo del programa es sumamente sencillo, ya que sólo se compone de la apertura de ficheros, proceso, cierre y listado.

PROGRAMA LISTA POR FECHAS 11



STE programa sirve para obtener del fichero de movimientos (Apuntes) especificando por el usuario una lista de movimientos de una fecha determinada, o se detecte final de fichero.

Ejemplo de resultado de este programa:

Listado para fecha 10-10-86

Mov.	**Fecha**	**Origen**	**Destino**	**Concepto**	**Ptas.**
2	10/10/86	POSTAL	CAJA CASA	saldo inicial	80.000
3	10/10/86		AHORROS	saldo inicial	200.000
4	10/10/86		HISPANO	saldo inicial	50.000
5	10/10/86			saldo inicial	1.250.000



PROGRAMA FECHA, EN PASCAL

```
Program listafecha(output) ;
 Type
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;
 regdes = record
  tabdes : array[1..20] of coddes ;
     end ;
 regcon = record
  tabcon : array[1..20] of codcta ;
```

```
     end ;
  regdcon = record
   tabdcon : array[1..20] of coddes ;
     end ;

Var
   Tablacta   : file of regcta ;
   Tablades   : file of regdes ;
   Tablacon   : file of regcon ;
   Tabladcon  : file of regdcon ;
   tabctareg   : regcta ;
   tabdesreg   : regdes ;
   tabconreg   : regcon ;
   tabdconreg  : regdcon ;
   Apuntes     : TEXT ;
   Listar      : Text ;
   sifcha,silista,otrafcha : boolean ;
   org : string[3] ;
   des : string[3] ;
   con : string[3] ;
   fchae   :  string[6] ;
   fchaf  : string[6] ;
   lista :  string[80] ;
   origen : string[15] ;
   destino : string[15] ;
   con1 : string[3] ;
   concepto :string[15] ;
   I,mov,c,chk,ic,id : Integer ;
   cifra : string[10] ;
   dd : string[2] ;
   mm : string[2] ;
   aa : string[2] ;
   a : string[1] ;

Procedure DIARIO ;
Var
 dia : integer ;
 sw  : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   ') ;
    writeln('Deme el dia...') ;
    writeln('  ') ;
    read(dia) ;
    writeln('  ') ;
   if (not(dia in[1..31])) then
     begin
```

```
      writeln('  ' ) ;
      writeln('no valido el dia....',dia:2) ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
 str(dia:2,dd) ;
 if dd[1] = ' ' then dd[1] := '0' ;
End ;

Procedure MENSUAL ;
Var
 mes : integer ;
 sw  : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   .') ;
    writeln('Deme el mes...') ;
    writeln('  ') ;
    read(mes) ;
    writeln('  ') ;
   if (not(mes in[1..12])) then
     begin
      writeln('  ' ) ;
      writeln(' no valido el mes ....',mes:2) ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
 str(mes:2,mm) ;
 if mm[1] = ' ' then mm[1] := '0' ;
End ;

Procedure ANUAL ;
Var
 anno : integer ;
 sw   : boolean ;
Begin
  sw := false ;
  Repeat
    writeln('   ') ;
    writeln('Deme el año...') ;
    writeln('  ') ;
    read(anno) ;
    writeln('  ') ;
```

```
   if (not(anno in[80..99])) then
     begin
      writeln('  ' ) ;
      writeln(' no valido el año ....',anno:2) ;
      sw := false ;
     end
     else
      sw := true ;
  Until sw = true ;
 str(anno:2,aa) ;
 fchae := concat(dd,mm,aa) ;
End ;

Procedure ANCHO ;
 VAR
  c : integer ;
 begin
  For c:= length(origen)+1 to 15 do
   begin
    origen := concat( origen,' ');
   end ;
  For c:= length(destino)+1 to 15 do
   begin
    destino := concat(destino,' ') ;
   end ;
  For c:= length(concepto)+1 to 15 do
   begin
    concepto := concat(concepto,' ') ;
   end
 end ;

Procedure BUSCA ;
 VAR
   cc : Integer ;
 begin
  For cc := 1 to 20 do
   begin
    if org = tabctareg.tabcta[cc]  then
         origen := tabdesreg.tabdes[cc] ;
   end ;
  For cc := 1 to 20 do
   begin
    if des = tabctareg.tabcta[cc] then
     destino := tabdesreg.tabdes[cc] ;
    end ;
  For cc := 1 to 20 do
   begin
    if con = tabconreg.tabcon[cc] then
```

```
      concepto := tabdconreg.tabdcon[cc] ;
    end
  end ;

 Procedure CIERRE;
    begin
     close (Apuntes) ; close (Listar) ;
    end ;

Procedure Empezar ;
 Begin
   if (fchae = fchaf) then
     begin
      sifcha := true ;
      silista := true ;
      BUSCA ;
      ANCHO ;
      writeln(listar,mov:3,'   ',dd:2,'/',mm:2,'/',aa:2,'   ',origen,' ',destino,'
       ',concepto,' ',cifra:9 );
     end
  End ;


 Procedure LEER ;
  VAR
     II : real ;
     cc : integer ;
     line : string[35] ;
     cantidad :string[10] ;
  Begin
     Readln(Apuntes,line) ;
     mov := (mov+1) ;
     cantidad := copy(line,20,10) ;
     cc := pos('.',cantidad);
     cifra := copy(line,(20+cc),(10-cc));
     val(cifra,II,chk) ;
     if (not (chk = 0 ) ) then
       begin
        writeln('ERROR numero al convertir ');
        CIERRE
        end ;
     org := copy(line,8,3) ;
     des := copy(line,12,3) ;
     con1 := copy(line,16,3) ;
     fchaf := copy(line,1,6) ;
     con := con1 ;
      if con1[3] = ' ' then
       begin
```

```
        con := copy(con1,1,2);
        con := concat(con,' ') ;
      end ;
  End ;

 Procedure PROCESO ;
 Begin
  assign(apuntes,'Apuntes.mov') ; reset(Apuntes) ;
  assign(Listar,'Listar.cta') ; rewrite(Listar) ;
  while not(eof(apuntes)) do
    begin
     LEER ;
     EMPEZAR ;
    end
 End ;


Procedure Listado ;
 Begin
  assign (Listar,'Listar.cta') ; reset (Listar);
  clrscr ;
  writeln( ' Listado para fecha  ....',dd:2,' / ',mm:2,' / ',aa:2 ) ;
  writeln( ' ------------------------------------ ') ;
  writeln(' ');
  writeln('MOV    FECHA         ORIGEN          DESTINO       CONCEPTO
  writeln('---   --------    --------------- ------------- -------------

   PESETAS   ');
  ----------- ');

   writeln(' ');

   WHILE not EOF(Listar) Do
     Begin
      readln(Listar,lista) ;
      writeln(lista) ;


     end ;
   close (Listar) ;


  End ;
   Procedure PIDEFCHA ;
    BEGIN
     DIARIO  ;
     MENSUAL ;
     ANUAL   ;
    END ;

     { **** Cuerpo del Programa **** }
```

```
BEGIN
 assign(tablacta,'tabla.cta') ; reset(tablacta) ;
 assign(tablades,'tabla.des') ; reset(tablades) ;
 assign(tablacon,'tabla.cto') ; reset(tablacon) ;
 assign(tabladcon,'tabla.dto') ; reset(tabladcon) ;
 read(tablacta,tabctareg) ; read(tablades,tabdesreg) ;
 read(tablacon,tabconreg) ; read(tabladcon,tabdconreg) ;
 close (tablacta) ; close (tablacon) ;
 close (tablades) ; close (tabladcon) ;
 otrafcha := false ;
 REPEAT
 mov := 0 ;
 sifcha := false ; silista := false ;
 clrscr ;
  Repeat
   PIDEFCHA ;
   PROCESO ;

 if sifcha = false then
  begin
   clrscr ;
   writeln('  ') ;
   writeln('  A T E N C I O N ' ) ;
   writeln('  ') ;
   writeln('no encontrado en el fichero de APUNTES, la fecha.. ',dd:2,'/',mm:2,
   '/',aa:2) ;
   writeln('  ' );
   writeln('¿ Quiere volver a empezar ? (s / n ) ') ;
   readln(a) ;
    if a = 's' then
     begin
      clrscr ;
      close(Apuntes ) ;
     end
    else
     begin
     sifcha := true ;
     silista := false ;
    end ;
  end ;
Until sifcha = true ;
CIERRE ;
if silista = true then LISTADO ;
writeln('   ') ;
writeln('  ') ;
writeln(' ¿ Quiere alguna otra fecha ? ( s / n ) )' );
writeln('  ') ;
readln(a) ;
```

```
    if a = 'n' then otrafcha := true ;
Until otrafcha = true ;

END.
```



EXPLICACION DEL PROGRAMA

Este programa consta de once procedimientos, cada uno de los cuales realiza una determinada misión.

Los procedimientos: **LEER**, **ANCHO**, **BUSCA**, **CIERRE** y **LISTADO** funcionan igual que en el caso anterior.

El procedimiento **PIDEFCHA**, como su nombre indica, tan sólo lo que hace es pedir la fecha a través de los procedimientos **DIARIO**, **MENSUAL** y **ANUAL**.



Procedimiento Empezar

Este forma el registro a grabar, cuando la fecha leída y la fecha pedida son iguales; entonces pone a verdadero las variables «sifcha y silista» (en ingles, «true»).



Procedimiento Proceso

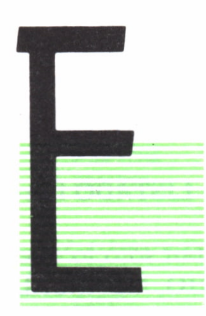
Abre los dos ficheros de Apuntes y Listar, controlando el fin del fichero de Apuntes.

En el cuerpo del programa, mediante la variable «otrafcha», definida como booleana, e inicializada a falso (en inglés, «false»), controlamos si después de una fecha queremos más o no.

Con la variable «sifcha», definida igual que la anterior, se controla si la fecha solicitada está o no en el fichero.

Por último, la variable «silista», igualmente definida, controla el listar o no el resultado.

PROGRAMA CONCEPTOS 12



STE programa sirve para obtener del fichero de movimientos (Apuntes) especificando por el usuario una lista de movimientos de un concepto determinado, o se detecte final de fichero.

Ejemplo de resultado de este programa:

Listado para el concepto . . . . .SAL

Mov.	Fecha	Origen	Destino	Concepto	Ptas.
1	10/10/86		BANESTO	saldo inicial	125.000
2	10/10/86		CAJA CASA	saldo inicial	80.000
3	10/10/86		AHORROS	saldo inicial	200.000
4	10/10/86		HISPANO	saldo inicial	50.000
5	10/10/86	POSTAL		saldo inicial	1.250.000



PROGRAMA CONCEPTOS, EN PASCAL

```
Program listaconcepto(output) ;
 Type
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;
 regdes = record
  tabdes : array[1..20] of coddes ;
     end ;
```

```
  regcon = record
   tabcon : array[1..20] of codcta ;
      end ;
  regdcon = record
   tabdcon : array[1..20] of coddes ;
      end ;

 Var
   Tablacta   : file of regcta ;
   Tablades   : file of regdes ;
   Tablacon   : file of regcon ;
   Tabladcon  : file of regdcon ;
   tabctareg   : regcta ;
   tabdesreg   : regdes ;
   tabconreg   : regcon ;
   tabdconreg  : regdcon ;
   Apuntes     : TEXT ;
   Listar      : Text ;
   sicon,silista : boolean ;
   org : string[3] ;
   des : string[3] ;
   con : string[3] ;
   sumat :  string[10];
   cantidad : string[10];
   cone   :  string[3]  ;
   lista :  string[80] ;
   origen : string[15] ;
   destino : string[15] ;
   con1 : string[3] ;
   concepto :string[15] ;
   suma,II : real ;
   I,mov,c,chk,ic,id : Integer ;
   cifra : string[10] ;
   dd : string[2] ;
   mm : string[2] ;
   aa : string[2] ;
   a : string[1] ;

   Procedure BUSCACON ;
    BEGIN
     REPEAT
      writeln('   ') ;
      writeln('Deme el concepto a buscar ') ;
      writeln('   ') ;
      read(cone) ;
       writeln('   ') ;
     For I := 1 to 20 do
       begin
```

```
        If tabconreg.tabcon[I] = cone then
         begin
          sicon := true ;
          id := I ;
         end ;
       end ;
       if sicon = false then writeln('Error. ',cone,' no encontrado en tabla ');
     UNTIL sicon = true ;
    END ;

Procedure ANCHO ;
 VAR
  c : integer ;
 begin
  For c:= length(origen)+1 to 15 do
   begin
    origen := concat( origen,' ');
   end ;
  For c:= length(destino)+1 to 15 do
   begin
    destino := concat(destino,' ') ;
   end ;
  For c:= length(concepto)+1 to 15 do
   begin
    concepto := concat(concepto,' ') ;
   end
 end ;

Procedure BUSCA ;
 VAR
   cc : Integer ;
 begin
  For cc := 1 to 20 do
   begin
    if org = tabctareg.tabcta[cc]  then
         origen := tabdesreg.tabdes[cc] ;
   end ;
  For cc := 1 to 20 do
   begin
    if des = tabctareg.tabcta[cc] then
     destino := tabdesreg.tabdes[cc] ;
    end ;
  For cc := 1 to 20 do
   begin
    if con = tabconreg.tabcon[cc] then
     concepto := tabdconreg.tabdcon[cc] ;
   end
 end ;
```

```
Procedure CIERRE;
   begin
    close (Apuntes) ; close (Listar) ;
   end ;


Procedure Empezar ;
 Begin
   if (cone = con1) then
     begin
      sicon := true ;
      silista := true ;
      BUSCA ;
      ANCHO ;
      writeln(listar,mov:3,'    ',dd:2,'/',mm:2,'/',aa:2,'    ',origen,' ',destino,
      ' ',concepto,' ',cifra:9 );
     end
  End ;


Procedure LEER ;
 VAR
    cc : integer ;
    line : string[35] ;
 Begin
    Readln(Apuntes,line) ;
    mov := (mov+1) ;
    cantidad := copy(line,20,10) ;
    cc := pos('.',cantidad);
    cifra := copy(line,(20+cc),(10-cc));
    val(cifra,II,chk) ;
    if (not (chk = 0 ) ) then
      begin
       writeln('ERROR numero al convertir ');
       CIERRE
       end ;
    org := copy(line,8,3) ;
    des := copy(line,12,3) ;
    con1 := copy(line,16,3) ;
    dd := copy(line,1,2) ;
    mm := copy(line,3,2) ;
    aa := copy(line,5,2) ;
    con := con1 ;
     if con1[3] = ' ' then
      begin
        con := copy(con1,1,2);
        con := concat(con,' ') ;
      end ;
 End ;
```

```
Procedure PROCESO ;
Begin
 assign(apuntes,'apuntes.mov') ; reset(apuntes) ;
 while not(eof(apuntes)) do
   begin
    LEER ;
    EMPEZAR ;
   end
End ;

Procedure Listado ;
 Begin
  assign (Listar,'Listar.cta') ; reset (Listar);
  clrscr ;
  writeln(' Listado para el concepto ....',tabconreg.tabcon[id] ) ;
  writeln( '  ') ;
  writeln('  ');
  writeln('MOV    FECHA         ORIGEN          DESTINO       CONCEPTO
  writeln('---   --------   --------------- -------------- --------------
   PESETAS   ');
  ---------- ');
   writeln('  ');

   WHILE not EOF(Listar) Do
     Begin
      readln(Listar,lista) ;
      writeln(lista) ;

     end ;
   close (Listar) ;

  End ;

     { **** Cuerpo del Programa **** }

BEGIN
 assign(tablacta,'tabla.cta') ; reset(tablacta) ;
 assign(tablades,'tabla.des') ; reset(tablades) ;
 assign(tablacon,'tabla.cto') ; reset(tablacon) ;
 assign(tabladcon,'tabla.dto') ; reset(tabladcon) ;
 assign(Listar,'Listar.cta') ; rewrite(Listar) ;
 mov := 0 ; suma := 0 ;
 read(tablacta,tabctareg) ; read(tablades,tabdesreg) ;
 read(tablacon,tabconreg) ; read(tabladcon,tabdconreg) ;
 close (tablacta) ; close (tablacon) ;
 close (tablades) ; close (tabladcon) ;
 sicon := false ; silista := false ;
 clrscr ;
```

```
Repeat
 iF sicon = false then
  begin
   BUSCACON ;
   sicon := false ;
  end ;
 PROCESO ;
 if sicon = false then
  begin
   clrscr ;
   writeln('  ') ;
   writeln('  A T E N C I O N ' ) ;
   writeln('  ') ;
   writeln('no encontrado en el fichero de APUNTES el codigo de concepto ',
   cone ) ;
   writeln('  ' );
   writeln('¿ Quiere volver a empezar ? (s / n ) ') ;
   readln(a) ;
    if a = 's' then
     begin
      clrscr ;
      close(Apuntes ) ;
     end
    else
     begin
     sicon := true ;
     silista := false ;
    end ;
  end ;
Until sicon = true ;
CIERRE ;
if silista = true then LISTADO ;
END.
```



EXPLICACION DEL PROGRAMA

Este programa consta de ocho procedimientos, cada uno de los cuales realiza una determinada misión.

Los procedimientos: PROCESO, LEER, ANCHO, BUSCA, CIERRE y LISTADO funcionan igual que en los casos anteriores.



Procedimiento Buscacon:

Este procedimiento lo único que hace es pedir el concepto a buscar dentro del fichero Apuntes, lo chequea con la tabla de conceptos, mediante las instrucciones siguientes:

```
For I: = 1 to 20 do
  begin
    If tabconreg.tabcon¿Iñ = cone then
      begin
        sicon: = true;
        id: = I;
      end;
    end;
```

En este conjunto de instrucciones podemos observar que si el concepto se encuentra dentro de la tabla cargada, pondrá a la variable «sicon» el valor de verdadero (en inglés «true»), valor que hace a este procedimiento salir del bucle («Repeat»), ya que en caso de no localizar el concepto en la tabla, la variable «sicon» tiene el valor de falso y hará dos cosas:

1. Escribirá mensaje de error con la instrucción siguiente:

```
if sicon = false then writeln ('Error. ',cone,' no encontrado en tabla');
```

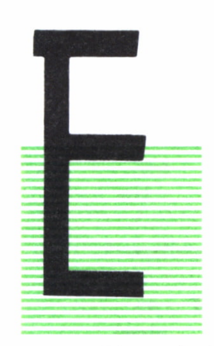
2. Como tiene valor de falso, no sale del bucle; esto lo controla la instrucción:

```
UNTIL sicon = true;
```

En el cuerpo del programa, mediante la variable «sicon», definida como booleana e inicializada a falso (en ingles «false»), controlamos si el concepto que se halla en la tabla también está en el fichero Apuntes, ya que en caso de no estar en el fichero, damos la opción de volver a empezar o salirnos.

Por último, la variable «silista» igualmente definida, controla el listar o no el resultado.

PROGRAMA RESUMEN DE UNA CUENTA 13



EN este programa el usuario define una cuenta, y el programa, después de chequear si la cuenta es o no correcta, lee secuencialmente todo el fichero, y escribe un informe como el siguiente contenido:

— Lista de los conceptos e importes de las entradas que ha tenido la cuenta, y un total de los mismos.
— Lista de los conceptos e importes de las salidas que ha tenido la cuenta, y su total.
— Por último, el saldo de la cuenta.

Una muestra de la salida podía ser la siguiente:

Resumen de la cuenta ....BANESTO

Entradas	**Importes Pesetas**	
Saldo inicial	128.000	
INGRESO	150.000	
Total entradas	278.000	
Salidas	**Importes**	
COLEGIO	15.000	
GAS	2.500	
Total salidas	18.000	
Saldo de la cuenta	BANESTO	260.000



PROGRAMA RESUMEN, EN PASCAL

```
Program RESUMEN(output) ;
 Type
```

```
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;
 regdes = record
  tabdes : array[1..20] of coddes ;
     end ;
 regcon = record
  tabcon : array[1..20] of codcta ;
     end ;
 regdcon = record
  tabdcon : array[1..20] of coddes ;
     end ;


Var
  Tablacta   : file of regcta ;
  Tablades   : file of regdes ;
  Tablacon   : file of regcon ;
  Tabladcon  : file of regdcon ;
  tabctareg   : regcta ;
  tabdesreg   : regdes ;
  tabconreg   : regcon ;
  tabdconreg  : regdcon ;
  Apuntes    : TEXT ;
  Nombre     : string[6] ;
  tabsumaP   : array[1..20] of real ;
  tabsumaA   : array[1..20] of real ;
  tconcep    : array[1..20] of string[15] ;
  sicta,mascta : boolean ;
  salinit : string[15] ;
  cifrinit : string[10] ;
  org : string[3] ;
  des : string[3] ;
  con : string[3] ;
  cantidad : string[10];
  cta   :  string[3]  ;
  lista :  string[6] ;
  origen : string[15] ;
  destino : string[15] ;
  con1 : string[3] ;
  concepto :string[15] ;
  II,III : real ;
  I,c,chk,ic,emp,id : Integer ;
  cifra : string[10] ;
  a : string[1] ;
```

```
Procedure ANCHO ;
 VAR
  c : integer ;
 begin
  For c:= length(origen)+1 to 15 do
   begin
    origen := concat( origen,' ');
   end ;
  For c:= length(destino)+1 to 15 do
   begin
    destino := concat(destino,' ') ;
   end ;
  For c:= length(concepto)+1 to 15 do
   begin
    concepto := concat(concepto,' ') ;
   end
 end ;

Procedure INICIALIZA ;
 VAR
   cc : Integer ;
 begin
  For cc := 1 to 20 do
   begin
    tabsumaP[cc] := 0 ;
    tabsumaA[cc] := 0 ;
   end ;
 end ;

Procedure BUSCA ;
 VAR
   cc : Integer ;
 begin
  sicta := true ;
  For cc := 1 to 20 do
  begin
    if con = tabconreg.tabcon[cc] then
     begin
     concepto := tabdconreg.tabdcon[cc] ;
     ic := cc ;
     end ;
  end ;
    if con = 'SAL' then
      begin
       salinit  := concepto ;
       cifrinit := cantidad ;
       lista := 'Pasivo' ;
        if des = cta  then  lista := 'Activo' ;
```

```
      end
     else
      begin
       if nombre = 'Pasivo' then
       tabsumaP[ic] := tabsumaP[ic] + II ;
       if nombre = 'Activo' then
       tabsumaA[ic] := tabsumaA[ic] + II ;
      end ;
End;

      Procedure CIERRE;
         begin
          close (Apuntes) ;
         end ;


      Procedure Nosal ;
       begin
         if (org = cta ) then
           begin
            nombre := 'Pasivo' ;
            BUSCA ;
            ANCHO ;
           end ;
          if (des = cta ) then
            begin
             nombre := 'Activo' ;
             BUSCA ;
             ANCHO ;
            end ;
         end ;


      Procedure Empezar ;
       Begin
         if (org = cta) or (des = cta) then
           begin
            if emp = 1 then nosal ;
             if (con = 'SAL' ) then
              begin
               BUSCA ;
               ANCHO ;
               emp := 1 ;
              end ;
            end
        End ;
```

```
Procedure LEER ;
 VAR
    cc : integer ;
    line : string[35] ;
 Begin
    Readln(Apuntes,line) ;
    cantidad := copy(line,20,10) ;
    cc := pos('.',cantidad);
    cifra := copy(line,(20+cc),(10-cc));
    val(cifra,II,chk) ;
    if (not (chk = 0 ) ) then
      begin
       writeln('ERROR numero al convertir ');
       CIERRE
       end ;
    org := copy(line,8,3) ;
    des := copy(line,12,3) ;
    con1 := copy(line,16,3) ;
    con := con1 ;
     if con1[3] = ' ' then
      begin
        con := copy(con1,1,2);
        con := concat(con,' ') ;
      end ;
 End ;

 Procedure PROCESO ;

 Begin
  assign(Apuntes,'apuntes.mov') ; reset(Apuntes) ;
  while not(eof(apuntes)) do
    begin
     LEER ;
     EMPEZAR ;
    end
 End ;

 Procedure Listado ;
 Var
   cc : integer ;
   emp : boolean ;
   cifra : string[10] ;
   suma : real ;
 Begin
   clrscr ;
   writeln( 'Resumen de la cuenta ....',tabdesreg.tabdes[id] ) ;
   writeln( '--------------------') ;
   writeln('  ');
   writeln('   Entradas                        Importes  ');
```

```
writeln('-------------------          ----------');
II := 0 ;  suma := 0 ;
if lista = 'Activo' then
 begin
   cc := pos('.',cifrinit);
   cifra := copy(cifrinit,(1+cc),(10-cc));
   val(cifra,III,chk) ;
   if (not (chk = 0 ) ) then
      writeln('ERROR numero al convertir ');
   II := II + III ;
   writeln(salinit,'              ',cifra:10 ) ;
 end ;
 cc := 1 ;
 emp := false ;

 Repeat
  if not (tabsumaA[cc] = 0) then
   begin
    str(tabsumaA[cc]:10:0,cifra) ;
    II := II + tabsumaA[cc] ;
    writeln(tabdconreg.tabdcon[cc],'              ',cifra:10);
   end ;
   cc := cc + 1 ;
   if cc > 20 then emp := true ;
 Until emp = true ;

 str(II:10:0,cifra);
 writeln( '                             ---------- ' );
 writeln('Total Entradas..............',cifra:10) ;
 writeln('  ');
 suma := suma + II;
 writeln('   Salidas                    Importes  ');
 writeln('-------------------          ----------');
 II := 0 ;
 if lista = 'Pasivo' then
  begin
   cc := pos('.',cifrinit);
   cifra := copy(cifrinit,(1+cc),(10-cc));
   val(cifra,III,chk) ;
   if (not (chk = 0 ) ) then
    writeln('ERROR numero al convertir ');
   II := II + III ;
    writeln(salinit,'                 ',cifra) ;
 end ;
 cc := 1 ;
 emp := false ;
Repeat
```

```
    if not (tabsumaP[cc] = 0) then
     begin
    str(tabsumaP[cc]:10:0,cifra) ;
    II := II + tabsumap[cc] ;
    writeln(tabdconreg.tabdcon[cc],'                  ',cifra);
   end ;
   cc := cc + 1 ;
   if cc > 20 then emp := true ;
 Until emp = true ;

 str(II:10:0,cifra);
 writeln('                             -----------') ;
 writeln('Total Salidas................',cifra ) ;
 if lista = 'Activo' then suma := suma-II ;
 if lista = 'Pasivo' then suma := II - suma ;
 str(suma:10:0,cifra) ;
 writeln('   ') ;
 writeln('Saldo de la cuenta........ ',tabdesreg.tabdes[id],'   ',cifra:10);
 writeln('                                             ========== ');
END;
  { **** Cuerpo del Programa **** }

BEGIN
 assign(tablacta,'tabla.cta') ; reset(tablacta) ;
 assign(tablades,'tabla.des') ; reset(tablades) ;
 assign(tablacon,'tabla.cto') ; reset(tablacon) ;
 assign(tabladcon,'tabla.dto') ; reset(tabladcon) ;

 read(tablacta,tabctareg) ; read(tablades,tabdesreg) ;
 read(tablacon,tabconreg) ; read(tabladcon,tabdconreg) ;
 close (tablacta) ; close (tablacon) ;
 close (tablades) ; close (tabladcon) ;
 mascta := true ;
 Repeat
  emp := 0 ;  sicta := false ;
  INICIALIZA ;
  clrscr ;

  REPEAT
   Repeat
    writeln('   ') ;
    writeln('Deme la cuenta a buscar ') ;
    writeln('   ') ;
    read(cta) ;
    writeln('   ') ;
     if length(cta) < 3 then
      writeln('ERROR. en la longitud, teclee de nuevo la cuenta ') ;
   Until length(cta) = 3 ;
```

```
    For I := 1 to 20 do
     begin
      If tabctareg.tabcta[I] = cta then
       begin
        sicta := true ;
        id := I ;
       end ;
     end ;
    if sicta = false then writeln('Error. ',cta:3,' no encontrada en tabla ');
    UNTIL sicta = true ;
    sicta := false ;
   PROCESO ;
   CIERRE ;
   if sicta = true then LISTADO ;
   if sicta = false then
    begin
     writeln(' ' ) ;
     writeln(' La cuenta ',cta,' no tiene movimiento en el fichero ') ;
    end ;
   Writeln(' ') ;
   writèln(' ') ;
   writeln(' ¿ Quiere alguna otra cuenta ? ( s / n ) ') ;
   readln(a) ;
    if a = 'n' then mascta := false ;
  Until mascta = false ;

END.
```



EXPLICACION DEL PROGRAMA

Este programa consta de ocho procedimientos, cada uno de los cuales realiza una determinada misión.

Los procedimientos: PROCESO, LEER, ANCHO y CIERRE funcionan igual que en los casos anteriores.



Procedimiento Inicializa

Pone a cero cada uno de los contadores de Activo y Pasivo, definidos como tablas, de igual número de filas que los conceptos existentes, en las tablas de conceptos; veamos sus instrucciones:

```
For cc := 1 to 20 do
 begin
  tabsumaP[cc] := 0 ;
  tabsumaA[cc] := 0 ;
 end ;
```



Procedimiento Busca

— Primero localiza el concepto leído en el fichero de Apuntes, con la tabla de conceptos, y guarda el índice que ocupa dicho concepto en la tabla, asignando a la variable «conceptos» la descripción de la tabla:

```
For cc := 1 to 20 do
begin
  if con = tabconreg.tabcon[cc] then
   begin
   concepto := tabdconreg.tabdcon[cc] ;
   ic := cc ;
   end ;
end ;
```

— Si es «saldo inicial», copia los datos necesarios, y dependiendo que la cuenta sea destino u origen, asignamos a la variable «lista» el valor de «activo o pasivo»:

```
if con = 'SAL' then
  begin
   salinit  := concepto ;
   cifrinit := cantidad ;
   lista := 'Pasivo' ;
    if des = cta  then  lista := 'Activo' ;
  end
```

— Si no es «saldo inicial», según el contenido de la variable «nombre», sea activo o pasivo, sumaremos el valor correspondiente en su tabla.

```
if con = 'SAL' then
  begin
    ...................
    ...................
    ...................
  end
 else
  begin
    if nombre = 'Pasivo' then
    tabsumaP[ic] := tabsumaP[ic] + II ;
    if nombre = 'Activo' then
    tabsumaA[ic] := tabsumaA[ic] + II ;
  end ;
```

Procedimiento Empezar

Aquí si alguna de las cuentas de origen o destino es igual a la cuenta pedida, quiere esto decir que es un registro bueno, en cuyo caso se procede a buscar si el concepto es «saldo inicial». Caso de serlo, se llama al procedimiento BUSCA Y ANCHO, activando la variable «emp» con el valor de 1, con lo cual al volver a entrar en este procedimiento, hará que saltemos (bifurcar) al procedimiento NOSAL.

Veamos las instrucciones:

```
if (org = cta) or (des = cta) then
  begin
    if emp = 1 then nosal ;
     if (con = 'SAL' ) then
      begin
        BUSCA ;
        ANCHO ;
        emp := 1 ;
      end ;
    end
```

Procedimiento NOSAL

En este procedimiento, si la cuenta origen es igual a la cuenta solicitada, asignamos a la variable (nombre) los caracteres de “Pasivo”, proce-

diendo a llamar a continuación a los procedimientos de Busca y Ancho. Veamos las instrucciones:

```
if (org = cta ) then
  begin
   nombre := 'Pasivo' ;
   BUSCA ;
   ANCHO ;
  end ;
```

Si por el contrario, la cuenta destino es la que es igual a la cuenta solicitada, asignaremos a la variable (nombre) los caracteres de "Activo", y como en el caso de la cuenta origen, también llamaremos a Busca y Ancho. Veamos las instrucciones:

```
if (des = cta ) then
  begin
   nombre := 'Activo' ;
   BUSCA ;
   ANCHO ;
  end ;
```



Procedimiento Listado

— Controla si el contenido de la variable «lista» tiene «Activo o Pasivo» para escribir el saldo inicial en la parte de entradas o en salidas. Veamos las instrucciones:

```
if lista = 'Activo' then
 begin
   cc := pos('.',cifrinit);
   cifra := copy(cifrinit,(1+cc),(10-cc));
   val(cifra,III,chk) ;
   if (not (chk = 0 ) ) then
      writeln('ERROR numero al convertir ');
   II := II + III ;
   writeln(salinit,'                    ',cifra:10 ) ;
 end ;
```

— Después, mientras las tablas donde hemos almacenado los importes tengan valor distinto de cero y no se llegue al final, se escribirá su contenido; esto se hace con las instrucciones siguientes:

```
cc := 1 ;
emp := false ;

Repeat
 if not (tabsumaA[cc] = 0) then
  begin
   str(tabsumaA[cc]:10:0,cifra) ;
   II := II + tabsumaA[cc] ;
   writeln(tabdconreg.tabdcon[cc],'       ',cifra:10);
  end ;
  cc := cc + 1 ;
  if cc > 20 then emp := true ;
Until emp = true ;
```

— Igual sería para la tabla de Pasivo.

En el cuerpo del programa controlamos que la cuenta a buscar esté en la tabla o nos dé un mensaje de error.

```
For I := 1 to 20 do
 begin
  If tabctareg.tabcta[I] = cta then
   begin
    sicta := true ;
    id := I ;
   end ;
 end ;
if sicta = false then writeln('Error. ',cta:3,' no encontrada en tabla ');
```

Controlamos que la cuenta a buscar tenga la longitud adecuada.

```
Repeat
 writeln('   ') ;
 writeln('Deme la cuenta a buscar ') ;
 writeln('   ') ;
 read(cta) ;
 writeln('   ') ;
  if length(cta) < 3 then
   writeln('ERROR. en la longitud, teclee de nuevo la cuenta ') ;
Until length(cta) = 3 ;
```

Si la variable «sicta» tiene valor verdadero, ejecutará el procedimiento Listado.

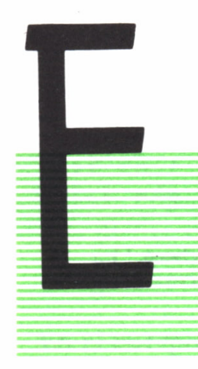
```
if sicta = true then LISTADO ;
```

Si su contenido es falso, entonces nos escribirá un mensaje. Veamos sus instrucciones:

```
mascta := true ;
Repeat

 if sicta = false then
  begin
   writeln('  ') ;
   writeln(' La cuenta  ',cta,' no tiene movimiento en el fichero ') ;
  end ;
 writeln('  ') ;
 writeln('  ') ;
 writeln(' ¿ Quiere alguna otra cuenta  ? ( s / n ) ') ;
 readln(a) ;
 if a = 'n' then mascta := false ;
Until mascta = false ;
```

PROGRAMA CIERRE DE CUENTAS 14



STE programa realiza lo siguiente:

Pide la fecha de cierre de cuentas, mediante los procedimientos: DIARIO, MENSUAL y ANUAL.

— Asigna a la variable (cta.) la primera cuenta de la tabla, escribiendo en la pantalla «Cierre de la cuenta... BEC»; a continuación se llama con el procedimiento Proceso. Veamos sus instrucciones:

```
cta :=  tabctareg.tabcta[ID] ;
if not (cta = '   ') then
 begin
   writeln('   ');
   writeln(' Cierre de la cuenta .... ',cta);
   writeln('   ');
   ID := ID + 1 ;
   nocta := true ;
   PROCESO ;
   .................
   .................
   .................
```

En el procedimiento Proceso empezamos cerrando el fichero Apuntes, y lo volvemos a abrir con objeto de situar el puntero al principio del fichero; esto se hace por cada cuenta de la tabla, leyéndose todo el fichero y efectuándose el cálculo del saldo de la cuenta. Veamos sus instrucciones:

```
close(Apuntes)  ;
assign(Apuntes,'apuntes.mov') ;
reset(Apuntes) ;
tabsuma := 0  ;
emp := 0 ;
while not(eof(apuntes)) do
  begin
   LEER ;
   EMPEZAR ;
  end
```

— Los procedimientos Leer y Empezar no tienen diferencia con los hasta ahora expuestos.
— El procedimiento Busca, que es llamado por los procedimientos Empezar y Nosal (según que sea el registro saldo inicial o no), activa para cada cuenta, en las variables (cuenta y nombre), la combinación de:

  «Origen y Pasivo»
  «Destino y Activo»

  claves que nos servirán para ir formando el saldo correcto de la cuenta que tratemos. Veamos las instrucciones:

Asignación:

```
if con = 'SAL' then
 begin
  nombre := 'Pasivo' ;
  cuenta := 'origen ';
   if des = cta  then
    begin
     cuenta := 'destino' ;
     nombre := 'Activo' ;
    end ;
 end ;
```

Después de la asignación, efectúa la suma aritmética en el contador, dependiendo de la asignación hecha anteriormente. Veamos sus instrucciones:

```
if (nombre = 'Pasivo') and (cuenta = 'destino') then
 tabsuma := tabsuma - II ;
```

```
if (nombre = 'Pasivo') and (cuenta = 'origen ') then
  tabsuma := tabsuma + II ;
if (nombre = 'Activo') and ( cuenta = 'destino') then
  tabsuma := tabsuma + II ;
if (nombre = 'Activo') and (cuenta = 'origen ' ) then
  tabsuma := tabsuma - II ;
```

— Escribe en un fichero provisional los datos de la cuenta leída y calculada, que, dependiendo sea una cuenta de origen o de destino, pondrá como saldo inicial para nueva cuenta dicha cuenta. Veamos sus instrucciones:

```
  suma := tabsuma ;
  org := cta  ;
  des := '   ' ;
  con := 'SAL' ;

  if cuenta = 'destino' then
   begin
    des := cta  ;
    org := '   ' ;
    suma := tabsuma ;
   end ;

   .....................
   .....................
   .....................

 writeln(Balance,fcha,' ',org,' ',des,' ',con,' ',cifra ) ;
```

— Conserva el criterio de la cuenta, es decir, si una cuenta era originalmente de origen después del cierre, sigue siendo de origen, aunque el resultado del cálculo del saldo sea negativo.
— Si una cuenta no se encuentra en el fichero, lo detecta y escribe un mensaje como éste:

La cuenta BEC no se encuentra en el fichero

— Pide un nombre para guardar el fichero de Apuntes antiguo, lo renombra con el nombre dado.

Al provisional entonces le da el nombre primitivo de Apuntes.mov.



Programa Cierre, en Pascal

```
Program cierre(output) ;
 Type
   codcta = string[3] ;
   coddes = string[15] ;
 regcta = record
  tabcta : array[1..20] of codcta ;
     end ;

 regcon = record
  tabcon : array[1..20] of codcta ;
     end ;
 Var
  Tablacta   : file of regcta ;
  Tablacon   : file of regcon ;
  tabctareg   : regcta ;
  tabconreg   : regcon ;
  BALANCE     : TEXT ;
  Apuntes     : TEXT ;
  Nombre      : string[6] ;
  titulo   :  string[8] ;
  titulof  :  string[12] ;
  sicta,nocta : boolean ;
  org : string[3] ;
  des : string[3] ;
  con : string[3] ;
  cantidad : string[10];
  cta   :  string[3]  ;
  lista :  string[6] ;
  con1 : string[3] ;
  II,tabsuma : real ;
  I,c,chk,ic,emp,id : Integer ;
  cifra : string[10] ;
  dd : string[2] ;
  mm : string[2] ;
  aa : string[2] ;
  fcha : string[6] ;
  cuenta : string[7] ;

Procedure DIARIO ;
 Var
  dia : integer ;
  sw  : boolean ;
 Begin
   sw := false ;
```

```
   Repeat
     writeln('   ') ;
     writeln('Deme el dia de cierre de cuentas') ;
     writeln('  ') ;
     read(dia) ;
     writeln('  ') ;
    if (not(dia in[1..31])) then
      begin
       writeln('  ' ) ;
       writeln('no valido el dia....',dia:2) ;
       sw := false ;
      end
      else
       sw := true ;
   Until sw = true ;
  str(dia:2,dd) ;
  if dd[1] = ' ' then dd[1] := '0' ;
 End ;

 Procedure MENSUAL ;
 Var
  mes : integer ;
  sw  : boolean ;
 Begin
   sw := false ;
   Repeat
     writeln('   ') ;
     writeln('Deme el mes...') ;
     writeln('  ') ;
     read(mes) ;
     writeln('  ') ;
    if (not(mes in[1..12])) then
      begin
       writeln('  ' ) ;
       writeln(' no valido el mes ....',mes:2) ;
       sw := false ;
      end
      else
       sw := true ;
   Until sw = true ;
  str(mes:2,mm) ;
  if mm[1] = ' ' then mm[1] := '0' ;
 End ;

 Procedure ANUAL ;
 Var
  anno : integer ;
  sw   : boolean ;
 Begin
```

```
    sw := false ;
    Repeat
      writeln('   ') ;
      writeln('Deme el año...') ;
      writeln('  ') ;
      read(anno) ;
      writeln('  ') ;
     if (not(anno in[80..99])) then
       begin
        writeln('  ' ) ;
        writeln(' no valido el año ....',anno:2) ;
        sw := false ;
       end
       else
        sw := true ;
    Until sw = true ;
   str(anno:2,aa) ;
   fcha := concat(dd,mm,aa) ;
  End ;

 Procedure BUSCA ;
  VAR
    cc : Integer ;
  begin
  nocta := false ;
   if con = 'SAL' then
    begin
     nombre := 'Pasivo' ;
     cuenta := 'origen ';
      if des = cta  then
       begin
        cuenta := 'destino' ;
        nombre := 'Activo' ;
       end ;
    end ;
  if (nombre = 'Pasivo') and (cuenta = 'destino') then
   tabsuma := tabsuma - II ;
  if (nombre = 'Pasivo') and (cuenta = 'origen ') then
   tabsuma := tabsuma + II ;
  if (nombre = 'Activo') and ( cuenta = 'destino') then
   tabsuma := tabsuma + II ;
  if (nombre = 'Activo') and (cuenta = 'origen ' ) then
   tabsuma := tabsuma - II ;

  End;

 Procedure CIERRE;
   begin
```

```
    close (Apuntes) ;
    close (Balance) ;
   end ;

Procedure Nosal ;
 begin
   if (org = cta ) then
     begin
      nombre := 'Pasivo' ;
      BUSCA ;
     end ;
    if (des = cta ) then
      begin
       nombre := 'Activo' ;
       BUSCA ;
      end ;
   end ;

Procedure Empezar ;
 Begin
   if (org = cta) or (des = cta) then
     begin
      if emp = 1 then nosal ;
       if (con = 'SAL' ) then
        begin
         BUSCA ;
         emp := 1 ;
        end ;
      end
  End ;


Procedure LEER ;
 VAR
    cc : integer ;
    line : string[35] ;
 Begin
    Readln(Apuntes,line) ;
    cantidad := copy(line,20,10) ;
    cc := pos('.',cantidad);
    cifra := copy(line,(20+cc),(10-cc));
    val(cifra,II,chk) ;
    if (not (chk = 0 ) ) then
      begin
       writeln('ERROR numero al convertir ');
       CIERRE
       end ;
    org := copy(line,8,3) ;
    des := copy(line,12,3) ;
```

```
    con1 := copy(line,16,3) ;
    con := con1 ;
     if con1[3] = ' ' then
      begin
        con := copy(con1,1,2);
        con := concat(con,' ') ;
      end ;
 End ;

Procedure GRABAR ;
Var
  cc,c : integer ;
  num : boolean ;
  cifra : string[10] ;
  suma : real ;
Begin
 suma := tabsuma ;
 org := cta  ;
 des := '   ' ;
 con := 'SAL' ;
 if cuenta = 'destino' then
   begin
    des := cta  ;
    org := '   ' ;
    suma := tabsuma ;
   end ;
 str(suma:10:0,cifra) ;
 num := false ;
  c := 2 ;
   Repeat
    if not(cifra[c] = ' ') then
     begin
      cc := c-1 ;
      cifra[cc] := '.' ;
      num := true ;
     end
     else
      c := c+1 ;
   Until num = true ;

   writeln(Balance,fcha,' ',org,' ',des,' ',con,' ',cifra ) ;

  END;

  Procedure PROCESO ;

  Begin
   close(Apuntes)  ;
```

```
  assign(Apuntes,'apuntes.mov') ;
  reset(Apuntes) ;
  tabsuma := 0  ;
  emp := 0 ;
  while not(eof(apuntes)) do
    begin
     LEER ;
     EMPEZAR ;
    end
 End ;

    { **** Cuerpo del Programa **** }

BEGIN
 assign(tablacta,'tabla.cta') ; reset(tablacta) ;
 assign(tablacon,'tabla.cto') ; reset(tablacon) ;
 read(tablacta,tabctareg) ;
 read(tablacon,tabconreg) ;
 close (tablacta) ; close (tablacon) ;
 assign(Balance,'Apun_tes.mov') ; rewrite(Balance) ;
  ID := 1 ;
  sicta := true ;
  clrscr ;

  DIARIO ;
  MENSUAL ;
  ANUAL ;
 Repeat
   cta :=  tabctareg.tabcta[ID] ;
   if not (cta = '    ') then
    begin
      writeln('   ');
      writeln(' Cierre de la cuenta .... ',cta);
      writeln('   ');
      ID := ID + 1 ;
      nocta := true ;
      PROCESO ;
       if nocta = false then
        GRABAR
       else writeln(' La cuenta ',cta,' no se encuentra en el fichero ') ;
    end
    else
     sicta := false ;
 Until sicta = false ;

  CIERRE ;

Repeat
  writeln('  ') ;
```

```
   writeln(' Deme el nombre con que quiere guardar el Fichero de Apuntes ');
   writeln(' del que hemos partido ');
   writeln(' ......máximo 8 caracteres .......');
   writeln('  ') ;
   readln(titulo) ;
   if length(titulo)<=0 then
     writeln('ERROR. debe teclear un nombre ')
    else
     sicta := true ;
Until sicta = true ;
   titulof := concat(titulo,('.mov'));
   rename(Apuntes,titulof) ;
   rename(Balance,'Apuntes.mov') ;

END.
```

APENDICE

El programa que se expone a continuación es la versión en BASIC del programa anteriormente visto en PASCAL.

Este programa está realizado para IBM pc,xt,at y compatibles, y puede ser utilizado por los usuarios de MSX que tengan unidad de disco, con sólo cambiar los argumentos de todas las sentencias LOCATE. Donde ponga LOCATE Y,X, X e Y pueden ser números o letras, los usuarios de MSX tendrán que poner LOCATE X,Y. O sea, invertir el orden de los argumentos.

Este programa, por su manera de trabajar y por utilizar varios archivos a la vez, no puede ser utilizado en aquellos ordenadores que carezcan de unidad de disco.

En caso de que usted tenga una unidad de disco en su Amstrad, Commodore o Spectrum, tendrá que hacer las modificaciones pertinentes en el programa en todo lugar donde se abran o cierren ficheros y se lean o se graben fichas.

```
1 REM CONTABILIDAD CASERA ---------> CONTA
2 REM Programa escrito por Fco. Morales y Adoracion Llena
3 REM (C) Ediciones Siglo Cultural, 1986
5 REM
6 KEY OFF
10 DIM TABCTA$(20),TABDES$(20),TABCPTO$(20),TABDCPTO$(20),TABSUMAA(20),TABSUMAP(
20)
15 REM
20 CLS:SCREEN 1:LINE(1,1)-(319,20),3,B:LINE(130,1)-(130,20),3
21 ORIGEN$="":DESTINO$="":CONCEPTO$="":PESETAS=0
23 LOCATE 2,2:PRINT "CONTABILIDAD":LOCATE 2,20:PRINT "MENU PRINCIPAL"
24 DATA CREACION DE TABLAS,SALDOS CUENTAS,ALTAS DE APUNTES,LISTADOS,RESUMEN GENE
RAL,RESUMEN POR FECHA,CIERRE,FIN
25 RESTORE 24:T=8:FOR I=1 TO T:READ A$:LOCATE 9+I,11:PRINT USING "##. Ñ
          Ñ";I,A$:NEXT
```

```
26 LOCATE 20,15:PRINT "ELIGE OPCION ";:INPUT OP:IF OP<1 OR OP>T THEN BEEP:BEEP:G
OTO 26
27 IF OP>3 THEN GOSUB 9800 ' lectura tablas
29 ON OP GOTO 100,1000,1400,1700,2200,2300,2400,2500
100 REM **********
110 REM * TABLAS *
120 REM **********
130 REM
150 OPEN "O",1,"TABLA.CTA"
160 OPEN "O",2,"TABLA.DES"
170 OPEN "O",3,"TABLA.CTO"
180 OPEN "O",4,"TABLA.DTO"
190 CLS
200 PRINT "Deme los codigos cta/concepto "
210 PRINT "y teclee END para salir del bucle"
220 PRINT
230 FOR I=1 TO 20
240 TABCTA$(I)=" ":TABDES$(I)=" ":TABCPTO$(I)=" ":TABDCPTO$(I)=" "
250 NEXT I
260 FOR I=1 TO 20
270 PRINT "Entre el codigo de cuenta ... (max 3 posiciones)"
280 INPUT CTA$
290 IF LEN(CTA$)>3 THEN GOTO 280
300 II=I-1
310 IF CTA$="END" THEN GOTO 390
320 TABCTA$(I)=CTA$
330 PRINT
340 PRINT "Entre la descripcion de la cuenta ... (max 15 posiciones)"
350 INPUT TABDES$(I)
360 IF LEN(TABDES$(I))>15 THEN GOTO 350
370 PRINT
380 NEXT I
390 CLS
400 FOR I=1 TO 20
410 PRINT "Entre el codigo del concepto ... (max 3 posiciones)"
420 INPUT CTA$
430 IF LEN(CTA$)>3 THEN GOTO 420
440 III=I-1
450 IF CTA$="END" THEN GOTO 530
460 TABCPTO$(I)=CTA$
470 PRINT
480 PRINT "Entre la descripcion del concepto ... (max 15 posiciones)"
490 INPUT TABDCPTO$(I)
500 IF LEN(TABDCPTO$(I))>15 THEN GOTO 490
510 PRINT
520 NEXT I
530 CLS
540 PRINT "TABLA DE CUENTAS Y DESCRIPCION"
550 PRINT "------------------------------"
```

```
560 PRINT:PRINT
570 PRINT "CODIGO CUENTA          DESCRIPCION"
580 PRINT "-------------          -----------"
590 PRINT
600 FOR I=1 TO II
610 PRINT "   ";TABCTA$(I);"           ";TABDES$(I)
620 NEXT I
630 PRINT "ESTA DE ACUERDO CON LA TABLA (S/N) "
640 GOSUB 20000
650 IF CO=0 THEN GOTO 690
660 FOR I=1 TO II:PRINT #1,TABCTA$(I):NEXT I
670 FOR I=1 TO II:PRINT #2,TABDES$(I):NEXT I
680 GOTO 710
690 PRINT "NO SE SALVA LA TABLA DE CUENTAS"
700 PRINT "ENTRELA DE NUEVO"
710 CLOSE#1:CLOSE#2
720 PRINT "Quiere continuar viendo la otra tabla? (s/n)"
730 GOSUB 20000
740 IF CO=0 THEN 845
750 CLS
760 PRINT "TABLA DE CONCEPTOS Y DESCRIPCIONES"
770 PRINT "----------------------------------"
780 PRINT :PRINT
790 PRINT "CODIGO CONCEPTO          DESCRIPCION"
800 PRINT "---------------          -----------"
810 PRINT
815 FOR I=1 TO II
820 PRINT "  ";TABCPTO$(I);"      ";TABDCPTO$(I)
825 NEXT I
830 PRINT:PRINT "ESTABA OK LA TABLA (S/N)":GOSUB 20000
835 IF CO=0 THEN GOTO 845
840 FOR I=1 TO III:PRINT #3,TABCPTO$(I):PRINT #4,TABDCPTO$(I):NEXT I
845 CLOSE#3:CLOSE#4
850 GOTO 20
1000 REM ****************
1010 REM * SALDO CUENTAS *
1020 REM ****************
1030 REM
1040 REM
1060 OPEN "I",1,"TABLA.CTA"
1070 OPEN "O",2,"APUNTES.MOV"
1080 CLS:PRINT "COMIENZA LA LECTURA DE TABLAS DE CUENTAS":I=0
1090 WHILE EOF(1)=0
1100 I=I+1
1110 INPUT#1,TABCTA$(I)
1120 WEND
1130 II=I
1140 PRINT :PRINT "DISPLAY DE LA TABLA DE CUENTAS LEIDA":PRINT
1150 FOR I=1 TO II:PRINT TABCTA$(I):NEXT I
```

```
1160 PRINT :PRINT " FIN DE LA LECTURA DE LA TABLA":PRINT
1165 GOSUB 21000
1170 CLOSE#1:CLS
1180 FOR A=1 TO II
1190 CLS:CTA$=TABCTA$(A)
1200 IF CTA$="" THEN GOTO 1370
1210 X=3:Y=1:GOSUB 9500 ' rutina fecha
1220 PTAS=0:ORG$="":DES$="":CON$="SAL"
1230 PRINT :PRINT "SALDO INICIAL PARA LA CUENTA .... ";CTA$:PRINT
1240 INPUT PTAS
1250 PRINT  "DIGAME SI ESTA CUENTA ES DE ORIGEN (1) O DE DESTINO (2)"
1260 PRINT :INPUT COT
1270 IF COT<1 OR COT >2 THEN BEEP:PRINT "ERROR.TECLEELO DE NUEVO":GOTO 1250
1280 IF COT=1 THEN ORG$=CTA$
1290 IF COT=2 THEN DES$=CTA$
1300 PRINT :PRINT "CONTENIDO DEL REGISTRO":PRINT
1310 PRINT USING "##/##/## ñ   ñ  ñ   ñ  ñ   ñ  #######";DD,MM,AA,ORG$,DES$,CON$
,PTAS
1320 PRINT :PRINT "¿ESTA DE ACUERDO? (S/N)":GOSUB 20000
1330 IF CO=0 THEN CLS:BEEP:GOTO 1210
1360 PRINT#2,FECHA$
1362 PRINT#2,ORG$
1363 PRINT#2,DES$
1364 PRINT#2,CON$
1365 PRINT#2,PTAS
1370 NEXT A
1380 CLOSE#2
1390 GOTO 20
1400 REM *********************
1403 REM * ALTAS DE APUNTES *
1405 REM *********************
1407 REM
1408 REM
1410 OPEN "I",1,"TABLA.CTA"
1420 OPEN "I",2,"TABLA.CTO"
1430 OPEN "I",3,"APUNTES.MOV"
1440 OPEN "O",4,"APTS NEW.MOV"
1450 WHILE EOF(3)=0
1455 INPUT #3,FECHA$,ORIGEN$,DESTINO$,CONCEPTO$,PESETAS
1460 PRINT#4,FECHA$
1461 PRINT#4,ORIGEN$
1462 PRINT#4,DESTINO$
1463 PRINT#4,CONCEPTO$
1464 PRINT#4,PESETAS
1465 WEND:CLOSE #3
1470 I=0:WHILE EOF(1)=0
1475 I=I+1:INPUT #1,TABCTA$(I)
1480 WEND:II=I
1485 I=0:WHILE EOF(2)=0
```

```
1490 I=I+1:INPUT #2,TABCON$(I)
1495 WEND:JJ=I
1499 CLOSE #1:CLOSE #2
1500 NOMBRE$="":ORG$="":DES$="":CON$="":PTAS=0
1510 CLS
1520 X=3:Y=1:GOSUB 9500 ' rutina fecha
1525 NOMBRE$="ORIGEN"
1530 PRINT :PRINT "DIGAME EL CODIGO DE LA CUENTA ";NOMBRE$
1535 X=1:Y=5:W$="A":M$="z":LO=3:GOSUB 9000:CTA$=D$
1545 SW=0:FOR A=1 TO II
1550 IF TABCTA$(A)=CTA$ THEN ORG$=CTA$:A=II:SW=1
1555 NEXT
1558 IF SW=0 THEN PRINT "ERROR, el codigo no esta en la tabla":PRINT "compruebel
o y tecleelo de nuevo":GOTO 1535
1559 Y=5:FOR I=1 TO 2:LOCATE Y+I,1:PRINT SPC(40);:NEXT
1560 PRINT :NOMBRE$="DESTINO"
1565 PRINT :PRINT "DIGAME EL CODIGO DE LA CUENTA ";NOMBRE$
1570 X=1:Y=11:GOSUB 9000:CTA$=D$
1580 SW=0:FOR A=1 TO II
1585 IF TABCTA$(A)=CTA$ THEN DES$=CTA$:A=II:SW=1
1590 NEXT
1591 IF SW=0 THEN PRINT "ERROR, el codigo no esta en la tabla":PRINT "compruebel
o y tecleelo de nuevo":GOTO 1570
1593 Y=11:FOR I=1 TO 2:LOCATE Y+I,1:PRINT SPC(40);:NEXT
1595 PRINT :PRINT "DEME EL CODIGO DEL CONCEPTO."
1600 X=1:Y=15:GOSUB 9000:CTA$=D$
1615 SW=0:FOR A=1 TO JJ
1620 IF TABCON$(A)=CTA$ THEN CON$=CTA$:A=JJ:SW=1
1625 NEXT
1626 IF SW=0 THEN PRINT "ERROR, el codigo no esta en la tabla":PRINT "compruebel
o y tecleelo de nuevo":GOTO 1600
1628 Y=15:FOR I=1 TO 2:LOCATE Y+I,1:PRINT SPC(40);:NEXT
1630 PRINT :PRINT "DEME EL IMPORTE DEL MOVIMIENTO..."
1635 X=1:Y=20:W$="0":M$="9":LO=10:GOSUB 9000:PTAS=VAL(D$)
1640 PRINT#4,FECHA$
1641 PRINT#4,ORG$
1642 PRINT#4,DES$
1643 PRINT#4,CON$
1644 PRINT#4,PTAS
1650 PRINT :PRINT " QUIERE REGISTRAR MAS APUNTES? (S/N)":GOSUB 20000
1655 IF CO=1 THEN 1500
1660 CLOSE #4
1665 KILL "APUNTES.MOV"
1670 NAME "APTS NEW.MOV" AS "APUNTES.MOV"
1680 GOTO 20
1700 REM ******************
1701 REM * MENU LISTADOS *
1702 REM ******************
1710 CLS:LINE(1,1) (319,20),3,B:LINE(130,1)-(130,20),3
```

```
1715 LOCATE 2,2:PRINT "CONTABILIDAD":LOCATE 2,20:PRINT "LISTADO APUNTES"
1720 DATA LISTADO POR FECHA,LISTADO POR CUENTA,LISTADO POR CONCEPTO,LISTADO POR
MOVIMIENTOS,MENU PRINCIPAL
1725 RESTORE 1720:T=5:FOR I=1 TO T:READ A$
1728 LOCATE 10+I,11:PRINT USING "##. Ñ                              Ñ";I,A$:NEXT
1730 LOCATE 18,15:PRINT "ELIGE OPCION ";:INPUT OP
1735 IF OP<1 OR OP>T THEN BEEP:BEEP:GOTO 1730
1790 ON OP GOTO 1800,1900,2000,2100,3000
1800 REM ****************
1801 REM * LISTADO FECHA *
1802 REM ****************
1810 MOV=0:SIFCHA=1:SILISTA=1
1815 CLS
1820 X=1:Y=1:GOSUB 9500' rutina fecha
1825 GOSUB 4000 ' rutina proceso1
1830 IF SIFCHA=0 THEN 1860
1835 CLS:PRINT :PRINT "ATENCION":PRINT
1840 PRINT "No encontrada en el fichero  de APUNTES":PRINT USING "la fecha ##/##
/##";DD,MM,AA
1845 PRINT :PRINT " QUIERE VOLVER A EMPEZAR? (S/N)":GOSUB 20000
1850 IF CO=1 THEN CLS:CLOSE #1:GOTO 1820 ELSE SIFCHA=0:SILISTA=1
1860 IF SILISTA=0 THEN GOSUB 4200 :GOSUB 21000:WIDTH 40' rutina listado
1865 PRINT :PRINT :PRINT " QUIERE ALGUNA OTRA FECHA? (S/N)":GOSUB 20000
1870 IF CO=1 THEN 1810
1875 GOTO 1700
1900 REM *******************
1901 REM * LISTADO cuentas *
1902 REM *******************
1910 CLS:MOV=0:SUMA=0:EMP=0:SICTA=1
1915 PRINT :PRINT "DEME LA CUENTA A BUSCAR":PRINT
1920 X=1:Y=4:LO=3:W$="A":M$="z":GOSUB 9000:CTA$=D$
1923 FOR I=1 TO II:IF TABCTA$(I)=CTA$ THEN SICTA=0:ID=I
1925 NEXT
1930 IF SICTA=1 THEN PRINT "ERROR ",CTA$," no encontrada en la tabla":GOTO 1920
1935 GOSUB 4600 ' proceso2
1940 CLOSE #1:CLOSE #2
1960 IF SICTA=0 THEN GOSUB 4700:GOSUB 21000:WIDTH 40' rutina listado2
1965 PRINT :PRINT :PRINT " QUIERE ALGUNA OTRA CUENTA? (S/N)":GOSUB 20000
1970 IF CO=1 THEN 1910
1975 GOTO 1700
2000 REM *******************
2001 REM * LISTADO CONCEPTO *
2002 REM *******************
2010 SICON=1:SILISTA=1
2015 CLS
2020 IF SICON=1 THEN GOSUB 4300:SICON=1 ' rutina buscacon2
2025 GOSUB 4400 ' rutina proceso2
2030 IF SICON=0 THEN 2060
2035 CLS:PRINT :PRINT "ATENCION":PRINT
```

```
2040 PRINT "No encontrada en el fichero  de APUNTES":PRINT "el codigo de concept
o ",CONE$
2045 PRINT :PRINT " QUIERE VOLVER A EMPEZAR? (S/N)":GOSUB 20000
2050 IF CO=1 THEN CLS:CLOSE #1:GOTO 2020 ELSE SICON=1:SILISTA=1
2060 IF SILISTA=0 THEN GOSUB 4500:GOSUB 21000:WIDTH 40' rutina listado2
2065 PRINT :PRINT :PRINT " QUIERE ALGUN OTRO CONCEPTO? (S/N)":GOSUB 20000
2070 IF CO=1 THEN 2010
2075 GOTO 1700
2100 REM ********************
2101 REM * LIS. POR MOVIM.  *
2102 REM ********************
2110 CLS:NMOV=0:PRIMERO=1:SIMOV=0
2115 GOSUB 4800' PROCESO4
2120 CLOSE #1:CLOSE#2
2125 GOSUB 4900:GOSUB 21000:WIDTH 40' rutina LISTADO4
2130 GOTO 1700
2200 REM ******************
2201 REM *  , RESUMEN      *
2203 REM ******************
2210 SICTA=1:EMP=0:NOMBRE$="":LISTA$=""
2220 FOR I=1 TO 20:TABSUMAP(I)=0:TABSUMAA(I)=0:NEXT
2225 CLS:PRINT :PRINT "DEME LA CUENTA A BUSCAR."
2230 X=1:Y=3:LO=3:W1$="A":M$="z":GOSUB 9000:IF LEN(D$)<3 THEN BEEP:GOTO 2230
2235 CTA$=D$
2240 FOR I=1 TO II
2245 IF TABCTA$(I)=CTA$ THEN ID=I:SICTA=0
2250 NEXT
2255 IF SICTA=1 THEN PRINT "ERROR, ",CTA$," no encontrada en la tabla.":GOTO 223
0
2260 SICTA=1
2265 GOSUB 6000 ' procesoR
2270 CLOSE #1
2275 IF SICTA=0 THEN WIDTH 80:GOSUB 6500:GOSUB 21000:WIDTH 40:GOTO 2285' listado
R
2280 PRINT :PRINT "La cuenta ",CTA$," no tiene movimientos."
2285 PRINT :PRINT " Quiere alguna otra cuenta? (S/N)":GOSUB 20000
2290 IF CO=1 THEN 2210
2295 GOTO 20
2300 REM ******************
2301 REM * RESUMEN FECHA  *
2303 REM ******************
2315 SICTA=1:EMP=0:SIFCHA=0:NOMBRE$="":LISTA$=""
2320 FOR I=1 TO 20:TABSUMAP(I)=0:TABSUMAA(I)=0:NEXT
2325 CLS:PRINT :PRINT "DEME LA CUENTA A BUSCAR."
2330 X=1:Y=3:LO=3:W1$="A":M$="z":GOSUB 9000:IF LEN(D$)<3 THEN BEEP:GOTO 2330
2335 CTA$=D$
2340 FOR I=1 TO II
2345 IF TABCTA$(I)=CTA$ THEN ID=I:SICTA=0
2350 NEXT
```

```
2355 IF SICTA=1 THEN PRINT "ERROR, ",CTA$," no encontrada en la tabla.":GOTO 233
0
2360 X=1:Y=6:GOSUB 9500:FECHA$=AA$+MM$+DD$' FECHA
2365 GOSUB 7000 ' procesoRF
2370 CLOSE #1
2375 IF SICTA=0 AND SIFCHA>0 THEN WIDTH 80:GOSUB 6500:GOSUB 21000:WIDTH 40:GOTO
2385' listadoR
2380 PRINT :PRINT "La cuenta ",CTA$," no tiene movimientos."
2385 PRINT :PRINT "¿Quiere alguna otra cuenta? (S/N)":GOSUB 20000
2390 IF CO=1 THEN 2315
2395 GOTO 20
2400 REM ******************
2401 REM *     CIERRE       *
2403 REM ******************
2413 OPEN "O",1,"APUN TES.MOV"
2415 SICTA=0:ID=1
2425 CLS:X=1:Y=2:GOSUB 9500
2430 CTA$=TABCTA$(ID)
2435 IF CTA$="" THEN 2465
2440 WIDTH 80:PRINT :PRINT "CIERRE DE LA CUENTA ... ",CTA$:PRINT
2445 ID=ID+1
2450 NOCTA=0
2453 GOSUB 8000
2455 IF NOCTA=1 THEN GOSUB 8400 ELSE PRINT "ERROR, ",CTA$," no encontrada en la
tabla."
2460 GOTO 2430
2465 CLOSE #1:GOSUB 21000
2470 CLS:PRINT :PRINT "Deme el nombre con el que quiere guardar el fichero de AP
UNTES del que hemos partido"
2475 PRINT "maximo 8 caracteres.":PRINT
2480 X=1:Y=7:LO=8:W$="A":M$="z":GOSUB 9000:TITULO$=D$+".MOV"
2485 NAME "APUNTES.MOV" AS TITULO$
2490 NAME "APUN TES.MOV" AS "APUNTES.MOV"
2495 GOTO 20
2500 RESET:SYSTEM
3000 GOTO 20
4000 ' RUTINA PROCESO1
4010 OPEN "I",1,"APUNTES.MOV"
4015 OPEN "O",2,"LISTAR.CTA"
4020 WHILE EOF(1)=0
4025 INPUT #1,FFECHA$,ORG$,DES$,CON$,PESETAS
4030 MOV=MOV+1
4035 IF FECHA$<>FFECHA$ THEN 4060
4040 SIFCHA=0:SILISTA=0
4045 GOSUB 4100 ' rutina busca
4055 GOSUB 5400 ' rutina grabacion
4060 WEND:CLOSE #1:CLOSE #2:RETURN
4100 ' RUTINA busca
4110 FOR I=1 TO II
```

```
4115 IF ORG$=TABCTA$(I) THEN ORIGEN$=TABDES$(I)
4120 IF DES$=TABCTA$(I) THEN DESTINO$=TABDES$(I)
4125 NEXT
4130 FOR I=1 TO JJ
4135 IF CON$=TABCON$(I) THEN CONCEPTO$=TABDCON$(I)
4140 NEXT
4150 RETURN
4200 ' RUTINA listado1
4205 WIDTH 80:OPEN "I",1,"LISTAR.CTA"
4210 PRINT USING "LISTADO PARA LA FECHA ##/##/##";DD,MM,AA
4215 PRINT "--------------------------------"
4220 GOSUB 5000 '  rutina cuerpo general
4225 CLOSE #1
4230 RETURN
4300 ' rutina buscacon
4305 PRINT :PRINT "DEME EL CONCEPTO A BUSCAR":PRINT
4310 X=1:Y=4:LO=3:W$="A":M$="z":GOSUB 9000:CONE$=D$:PRINT
4315 FOR I=1 TO JJ
4320 IF CONE$=TABCON$(I) THEN SICON=0:ID=I
4325 NEXT
4330 IF SICON=1 THEN PRINT "ERROR, ",CONE$," no encontrado en la tabla":GOTO 431
0
4340 RETURN
4400 ' RUTINA PROCESO3
4410 OPEN "I",1,"APUNTES.MOV"
4415 OPEN "O",2,"LISTAR.CTA"
4420 WHILE EOF(1)=0
4425 INPUT #1,FECHA$,ORG$,DES$,CON$,PESETAS
4430 MOV=MOV+1
4435 DD=VAL(LEFT$(FECHA$,2)):MM=VAL(MID$(FECHA$,3,2)):AA=VAL(RIGHT$(FECHA$,2))
4438 IF CONE$<>CON$ THEN 4460
4440 SICON=0:SILISTA=0
4445 GOSUB 4100 ' rutina busca
4455 GOSUB 5400 ' rutina grabacion
4460 WEND:CLOSE #1:CLOSE #2:RETURN
4500 ' RUTINA listado3
4505 WIDTH 80:OPEN "I",1,"LISTAR.CTA"
4510 PRINT "LISTADO PARA EL CONCEPTO ",TABCON$(ID)
4515 PRINT "------------------------------------"
4520 GOSUB 5000 '  rutina cuerpo general
4525 CLOSE #1
4530 RETURN
4600 ' RUTINA PROCESO2
4610 OPEN "I",1,"APUNTES.MOV"
4615 OPEN "O",2,"LISTAR.CTA"
4620 WHILE EOF(1)=0
4625 CON1$="":INPUT #1,FECHA$,ORG$,DES$,CON$,PESETAS
4630 MOV=MOV+1
4635 DD=VAL(LEFT$(FECHA$,2)):MM=VAL(MID$(FECHA$,3,2)):AA=VAL(RIGHT$(FECHA$,2))
```

```
4638 IF ORG$=CTA$ OR DES$=CTA$ THEN ELSE 4670
4640 IF EMP=1 THEN GOSUB 4680 ' NOSAL
4643 IF CON$<>"SAL" THEN 4670
4650 GOSUB 4100:CONCEPTO$="SALDO INICIAL"
4655 GOSUB 5400:PRINT #2,PESETAS
4660 SUMA=SUMA+PESETAS:EMP=1
4670 WEND:RETURN
4680 ' nosal
4685 IF ORG$=CTA$ THEN SUMA=SUMA-PESETAS:GOSUB 4100
4690 IF DES$=CTA$ THEN SUMA=SUMA+PESETAS:GOSUB 4100
4699 GOSUB 5400:PRINT #2,SUMA:RETURN
4700 ' RUTINA listado2
4705 WIDTH 80:OPEN "I",1,"LISTAR.CTA"
4710 PRINT "LISTADO PARA LA CUENTA ... ";TABDES$(ID)
4715 PRINT "-----------------------------------"
4720 GOSUB 5100 '  rutina cuerpo general
4725 CLOSE #1
4730 RETURN
4800 ' RUTINA PROCESO4
4810 OPEN "I",1,"APUNTES.MOV"
4815 OPEN "O",2,"LISTAR.CTA"
4820 CLS:PRINT :PRINT "DEME EL NO. DE MOVIMIENTO INICIAL"
4825 X=1:Y=4:LO=3:W$="0":M$="9":GOSUB 9000:MOVI=VAL(D$):MOVE=MOVI
4830 PRINT " CUANTOS MOVIMIENTOS QUIERE A PARTIR":PRINT USING "DEL MOVIMIENTO ##
#";MOVI:PRINT
4835 X=1:Y=8:GOSUB 9000:NUMMOV=VAL(D$)
4840 MOVF=MOVI+NUMMOV
4850 WHILE EOF(1)=0
4855 NMOV=NMOV+1:INPUT #1,FECHA$,ORG$,DES$,CON$,PESETAS
4860 IF MOVI=NMOV THEN SIMOV=0
4865 IF PRIMERO=0 THEN IF NMOV<=MOVF THEN MOVE=MOVE+1:SIMOV=0
4870 IF SIMOV=1 THEN 4895
4875 ORIGEN$="":DESTINO$="":CONCEPTO$="":GOSUB 4100' BUSCA
4880 DD=VAL(LEFT$(FECHA$,2)):MM=VAL(MID$(FECHA$,3,2)):AA=VAL(RIGHT$(FECHA$,2))
4885 MOV=MOVE:GOSUB 5400
4890 PRIMERO=0:SIMOV=1
4895 WEND:RETURN
4900 ' RUTINA listado4
4905 WIDTH 80:OPEN "I",1,"LISTAR.CTA"
4910 PRINT USING "LISTADO DE MOVIMIENTOS NOS.    ### AL ###";MOVI,MOVF-1
4915 PRINT "-------------------------------------"
4920 GOSUB 5000 '  rutina cuerpo general
4925 CLOSE #1
4930 RETURN
5000 '  RUTINA cuerpo general de listados
5010 PRINT "MOV     FECHA          ORIGEN              DESTINO             CONCEPTO
PESETAS"
5020 PRINT "---     -----          ------              -------             --------
-------"
```

```
5030 PRINT :WHILE EOF(1)=0
5040 INPUT #1,MOV,DD,MM,AA,ORG$,DES$,CON$,PESETAS
5050 PRINT USING "###  ##/##/##   Ñ              Ñ  Ñ              Ñ  Ñ
  Ñ #######";MOV,DD,MM,AA,ORG$,DES$,CON$,PESETAS
5060 WEND
5070 RETURN
5100 '  RUTINA cuerpo general de listado2
5110 PRINT "MOV   FECHA       ORIGEN        DESTINO        CONCEPTO       PTAS
  ACUMULADO"
5120 PRINT "---   -----       ------        -------        --------       ---
  ---------"
5130 PRINT :WHILE EOF(1)=0
5140 INPUT #1,MOV,DD,MM,AA,ORG$,DES$,CON$,PESETAS,SUMA
5150 PRINT USING "###  ##/##/## Ñ              Ñ Ñ              Ñ Ñ              Ñ
#######  ########";MOV,DD,MM,AA,ORG$,DES$,CON$,PESETAS,SUMA
5160 WEND
5170 RETURN
5400 'grabacion en LISTAR.cta
5410 PRINT #2,MOV
5420 PRINT #2,DD
5430 PRINT #2,MM
5440 PRINT #2,AA
5450 PRINT #2,ORIGEN$
5455 PRINT #2,DESTINO$
5460 PRINT #2,CONCEPTO$
5470 PRINT #2,PESETAS
5480 RETURN
6000 ' RUTINA procesoR
6010 OPEN "I",1,"APUNTES.MOV"
6020 WHILE EOF(1)=0
6025 INPUT #1,FECHA$,ORG$,DES$,CON$,PESETAS
6035 DD=VAL(LEFT$(FECHA$,2)):MM=VAL(MID$(FECHA$,3,2)):AA=VAL(RIGHT$(FECHA$,2))
6040 IF ORG$=CTA$ OR DES$=CTA$ THEN ELSE 6055
6045 IF EMP=1 THEN GOSUB 6100 ' nosal
6050 IF CON$="SAL" THEN GOSUB 6200:EMP=1 '  busca
6055 WEND
6060 RETURN
6100 ' RUTINA nosal
6110 IF ORG$=CTA$ THEN NOMBRE$="PASIVO":GOSUB 6200
6120 IF DES$=CTA$ THEN NOMBRE$="ACTIVO":GOSUB 6200
6130 RETURN
6200 ' RUTINA buscaR
6210 SICTA=0
6215 FOR I=1 TO JJ
6220 IF CON$=TABCON$(I) THEN CONCEPTO$=TABDCON$(I):IC=I
6225 NEXT
6230 IF CON$="SAL" THEN SALINIT$="SALDO INICIAL":CIFRINIT=PESETAS:LISTA$="PASIVO
":IF DES$=CTA$ THEN LISTA$="ACTIVO":GOTO 6245
6235 IF NOMBRE$="PASIVO" THEN TABSUMAP(IC)=TABSUMAP(IC)+PESETAS
```

```
6240 IF NOMBRE$="ACTIVO" THEN TABSUMAA(IC)=TABSUMAA(IC)+PESETAS
6245 RETURN
6500 ' RUTINA listado resumen
6505 CLS
6510 PRINT "RESUMEN DE LA CUENTA ... ",TABDES$(ID)
6515 PRINT "----------------------------"
6520 PRINT "      ENTRADAS                IMPORTES"
6525 PRINT "      --------                ---------"
6530 IX=0:SUMA=0
6535 IF LISTA$="ACTIVO" THEN IX=IX+CIFRINIT:PRINT USING "    ñ                ñ
    ########";SALINIT$,CIFRINIT
6540 I=1
6545 IF TABSUMAA(I)=0 THEN 6560
6550 IX=IX+TABSUMAA(I)
6555 PRINT USING "   ñ               ñ           ########";TABDCON$(I),TABSUMAA(I)
6560 I=I+1
6565 IF I<=20 THEN 6545
6570 PRINT :PRINT USING "TOTAL ENTRADAS ........  ##########";IX
6575 PRINT :PRINT :PRINT :SUMA=SUMA+IX
6580 PRINT "      SALIDAS                 IMPORTES"
6585 PRINT "      -------                 ---------"
6590 IX=0
6595 IF LISTA$="PASIVO" THEN IX=IX+CIFRINIT:PRINT USING "    ñ                ñ
    ########";SALINIT$,CIFRINIT
6600 I=1
6605 IF TABSUMAP(I)=0 THEN 6620
6610 IX=IX+TABSUMAP(I)
6615 PRINT USING "   ñ               ñ           ########";TABDCON$(I),TABSUMAP(I)
6620 I=I+1
6625 IF I<=20 THEN 6605
6630 PRINT :PRINT USING "TOTAL SALIDAS .........  ##########";IX
6635 IF LISTA$="ACTIVO" THEN SUMA=SUMA-IX
6640 IF LISTA$="PASIVO" THEN SUMA=IX-SUMA
6645 PRINT :PRINT USING "SALDO DE LA CUENTA ñ              ñ = #########";TABDES$(
ID),SUMA
6650 PRINT "                                  ========="
6660 RETURN
7000 ' RUTINA procesoRF
7010 OPEN "I",1,"APUNTES.MOV"
7020 WHILE EOF(1)=0
7025 INPUT #1,FECH$,ORG$,DES$,CON$,PESETAS
7030 A$=RIGHT$(FECH$,2):M$=MID$(FECH$,3,2):D$=LEFT$(FECH$,2):FECH$=A$+M$+D$
7040 IF ORG$=CTA$ OR DES$=CTA$ THEN ELSE 7055
7045 IF EMP=1 THEN GOSUB 7100
7050 IF CON$="SAL" THEN GOSUB 7200:EMP=1 '  busca
7055 WEND
7060 RETURN
7100 ' RUTINA nosal
7110 IF ORG$=CTA$ THEN NOMBRE$="PASIVO":GOSUB 7200
```

```
7120 IF DES$=CTA$ THEN NOMBRE$="ACTIVO":GOSUB 7200
7130 RETURN
7200 ' RUTINA buscaRF
7210 IF FECH$>FECHA$ THEN RETURN
7215 FOR I=1 TO JJ
7220 IF CON$=TABCON$(I) THEN CONCEPTO$=TABDCON$(I):IC=I
7225 NEXT
7230 IF CON$="SAL" THEN SALINIT$="SALDO INCIAL":CIFRINIT=PESETAS:LISTA$="PASIVO"
:IF DES$=CTA$ THEN LISTA$="ACTIVO":GOTO 7245
7235 IF NOMBRE$="PASIVO" AND FCHA$<=FECHA$ THEN TABSUMAP(IC)=TABSUMAP(IC)+PESETA
S
7240 IF NOMBRE$="ACTIVO" AND FCHA$<=FECHA$ THEN TABSUMAA(IC)=TABSUMAA(IC)+PESETA
S
7245 SIFCHA=SIFCHA+1
7250 RETURN
8000 ' RUTINA procesoC
8010 OPEN "I",2,"APUNTES.MOV"
8015 TABSUMA=0:EMP=0
8020 WHILE EOF(2)=0
8025 INPUT #2,FECH$,ORG$,DES$,CON$,PESETAS
8030 IF ORG$=CTA$ OR DES$=CTA$ THEN ELSE 8050
8035 IF EMP=1 THEN GOSUB 8100 ' NOSAL
8040 IF CON$="SAL" THEN GOSUB 8200:EMP=1
8050 WEND:CLOSE #2
8060 RETURN
8100 ' RUTINA nosal
8110 IF ORG$=CTA$ THEN NOMBRE$="PASIVO":GOSUB 8200
8120 IF DES$=CTA$ THEN NOMBRE$="ACTIVO":GOSUB 8200
8130 RETURN
8200 ' RUTINA buscaC
8220 NOCTA=1:IF CON$<>"SAL" THEN  8240
8225 NOMBRE$="PASIVO":CUENTA$="ORIGEN"
8230 IF DES$=CTA$ THEN CUENTA$="DESTINO":NOMBRE$="ACTIVO"
8240 IF (NOMBRE$="PASIVO" AND CUENTA$="DESTINO") OR (NOMBRE$="ACTIVO" AND CUENTA
$="DESTINO") THEN TABSUMA=TABSUMA+PESETAS
8250 IF (NOMBRE$="PASIVO" AND CUENTA$="ORIGEN") OR (NOMBRE$="ACTIVO" AND CUENTA$
="ORIGEN") THEN TABSUMA=TABSUMA-PESETAS
8260 RETURN
8400 ' RUTINA grabar
8410 SUMA=TABSUMA:ORG$=CTA$:DES$="":CON$="SAL"
8420 IF CUENTA$="DESTINO" THEN DES$=CTA$:ORG$=""
8430 PRINT #1,FECHA$
8431 PRINT #1,ORG$
8432 PRINT #1,DES$
8433 PRINT #1,CON$
8434 PRINT #1,SUMA
8440 RETURN
9000 REM ********************
9010 REM * ENTRADA DE DATOS *
```

```
9020 REM *********************
9030 REM
9040 LET DE=0:LET D$="":LOCATE Y,X:PRINT "_";
9050 FOR I=1 TO LO-1
9060    PRINT ".";
9070 NEXT I
9080 A$=INKEY$:IF A$="" THEN GOTO 9080
9090 IF ASC(A$)=8 AND LEN(D$)>0 THEN LOCATE Y,X-1 :PRINT "_.";:X=X-1:LET D$=LEFT
$(D$,LEN(D$)-1)
9100 IF A$="." AND DE=0 THEN LET DE=1:GOTO 9150
9110 IF A$="-" AND LEN(D$)=0 THEN GOTO 9150
9120 IF ASC(A$)=13 THEN GOTO 9170
9130 IF A$=" " THEN GOTO 9150
9140 IF A$>M$ OR A$<W$ THEN GOTO 9080
9150 D$=D$+A$:LOCATE Y,X:PRINT A$
9160 IF LEN(D$)<LO THEN X=X+1:LOCATE Y,X:PRINT "_":GOTO 9080
9170 RETURN
9500 REM *********************
9510 REM * ENTRADA DE FECHA *
9520 REM *********************
9530 REM
9540 XX=X:LOCATE Y,X:PRINT "ESCRIBA LA FECHA     dd/mm/aa"
9545 Y=Y+1
9550 LOCATE Y,X:PRINT ".. / .. / .."
9560 X=XX:LO=2:M$="9":W$="0":GOSUB 9000
9570 IF LEN(D$)<2 THEN 9560 ELSE DD$=D$:DD=VAL(D$)
9580 IF DD<1 OR DD>31 THEN GOTO 9560
9590 X=XX+5:GOSUB 9000
9600 IF LEN(D$)<2 THEN 9590 ELSE MM$=D$:MM=VAL(D$)
9610 IF MM<1 OR MM>12 THEN GOTO 9590
9620 X=XX+10:GOSUB 9000
9630 AA=VAL(D$):AA$=D$
9640 IF AA<80 OR AA>99 THEN GOTO 9620
9650 FECHA$=DD$+MM$+AA$
9660 RETURN
9800 REM ********************
9801 REM * Lectura TABLAS *
9802 REM ********************
9810 OPEN "I",1,"TABLA.CTA"
9815 OPEN "I",2,"TABLA.DES"
9820 OPEN "I",3,"TABLA.CTO"
9825 OPEN "I",4,"TABLA.DTO"
9830 I=0:WHILE EOF(1)=0
9835 I=I+1:INPUT #1,TABCTA$(I)
9840 WEND:II=I
9845 FOR I=1 TO II:INPUT #2,TABDES$(I):NEXT
9850 I=0:WHILE EOF(3)=0
9855 I=I+1:INPUT #3,TABCON$(I)
9860 WEND:JJ=I
```

```
7865 FOR I=1 TO JJ:INPUT #4,TABDCON$(I):NEXT
7870 CLOSE #1:CLOSE #2:CLOSE #3:CLOSE #4
7880 RETURN
20000 A$=INKEY$:IF A$="" THEN GOTO 20000
20010 D$="SsNn":IF INSTR(D$,A$)=0 THEN 20000
20020 D$="Ss":IF INSTR(D$,A$)=0 THEN CO=0 ELSE CO=1
20030 RETURN
21000 REM RUTINA PULSE UNA TECLA
21005 LOCATE 24,10:PRINT "PULSE UNA TECLA";
21010 A$=INPUT$(1):RETURN
```

La aparición de los ordenadores personales ha sido seguida por su impresionante proliferación, debida al abaratamiento de los materiales, lo que ha traído consigo el que se vaya introduciendo en el hogar, con la posibilidad de utilizarlo, tanto para distraer los ratos de ocio jugando, como para realizar otras aplicaciones de tipo práctico.

Nosotros, en este libro, nos vamos a referir al control doméstico y para ello utilizaremos un lenguaje sencillo para el lector. Al mismo tiempo, a lo largo del libro, se van introduciendo, además, conceptos básicos de contabilidad, que serán de utilidad al lector.